Securing Vehicular Communications - Assumptions, Requirements, and Principles

P. Papadimitratos
EPFL
Lausanne, Switzerland
panos.papadimitratos@epfl.ch

V. Gligor
University of Maryland
College Park, USA
gligor@eng.umd.edu

J-P. Hubaux
EPFL
Lausanne, Switzerland
jean-pierre.hubaux@epfl.ch

***Abstract*— Among civilian communication systems, vehicular networks emerge as one of the most convincing and yet most challenging instantiations of the mobile ad hoc networking technology. Towards the deployment of vehicular communication systems, security and privacy are critical factors and significant challenges to be met. Thanks to the substantial research efforts carried out by the community so far, we make the following contributions in this paper: we outline security requirements for vehicular communication systems, we provide models for the system and the communication, as well as models for the adversaries, and propose a set of design principles for future security and privacy solutions for vehicular communication systems.**

I. Introduction

Vehicular ad hoc networks (VANET) are a new technology that has recently drawn the attention of the industry and academia. *Vehicular communications (VC)* lie at the core of a number of research initiatives that aim to enhance safety and efficiency of transportation systems; with envisioned applications providing, for example, warnings on environmental hazards (e.g., ice on the pavement), traffic and road conditions (e.g., emergency braking, congestion, or construction sites), and local (e.g., tourist) information. In fact, vehicular networks emerge, among civilian communication systems, as one of the most convincing and yet most challenging instantiations of the mobile ad hoc networking technology.

To enable such applications, vehicles and road-side infrastructure units (RSUs), namely network nodes, will be equipped with on-board processing and wireless communication modules. Then, *vehicle-to-vehicle (V2V)* and *vehicle-to-infrastructure (V2I)* (bidirectional) communication will be possible directly when in range, or, in general, across multiple wireless links (hops), with nodes acting both as end points and routers. Relying on such hybrid networking appears to be the only means to realize safety and driving assistance applications, as an omnipresent infrastructure can be impractical, too costly, and thus very slowly deployed.

A comprehensive set of security mechanisms integrated into the VC systems is critical for their deployment. Otherwise, the efficiency of the transportation systems, as well as the physical safety of vehicles, drivers, and passengers could be jeopardized. Even worse, VC-based applications can be of life-critical nature. At the same time, VANETs are particulary challenging to secure due to the tight coupling between applications and the networking fabric, as well as additional societal, legal, and economical considerations, which raise a unique combination of operational and security requirements.

A small number of recent works are concerned with different aspects of security and privacy of vehicular networks, either outlining challenges [32], [26], describing particular attacks [19], [3] or more general attack overviews [1], offering general suggestions towards solutions [14], [29], or proposing mechanisms [15], [28], [21], [18]. Nevertheless, the literature provides neither a coherent view of the VC systems, with respect to their characteristics and the security and privacy requirements, nor a roadmap towards mechanisms that satisfy them.

In this paper, we seek to bridge this gap and provide a solid basis for the development of future vehicular security schemes. As VC is a technology in the making, our investigation draws from the current understanding and projections from both the academic and industry worlds. At the same time, we point out the unique or novel aspects due to VC salient characteristics.

In Sec. II, we first provide a concise problem statement and motivation and then list general security requirements. Schemes satisfying these requirements could be viewed as building blocks for any possible solution. In Sections III and IV we compile a set of operational characteristics and provide a minimal set of assumptions on the system and the communication model. Then, we investigate models of benign failures and models of adversarial behavior, and discuss the suitability of existing adversary models for the VC environment. In Sec. VI, we propose a set of design principles for any security solution for vehicular networks to follow. We conclude with a discussion on additional aspects and connections to practical considerations.

II. Security Requirements

The unique features of VC are a double-edged sword: a rich set of tools are offered to drivers and authorities (defined in Sec. III) but a formidable set of abuses and attacks becomes possible. Consider, for example, nodes that ’contaminate’ large portions of the vehicular network with false information: a single vehicle can transmit false hazard warnings (e.g., ice formation on the pavement), which can then be taken up by all vehicles in both traffic streams. Or, similarly, a vehicle that meaningfully modifies messages of other vehicles. Or even a vehicle that forges messages in order to masquerade an emergency vehicle to mislead other vehicles to slow down and yield.

These simple examples of exploits indicate that under all circumstances vehicular communications must be secured. In fact, it is possible that vehicles and their sensing, processing, and communication platforms are compromised. Worse even, any wireless-enabled device that runs a rogue version of the vehicular communication protocol stack poses a threat both to the vehicular network and the transportation system operation. Hence, the security of vehicular networks is indispensable; otherwise these systems could make anti-social and criminal behavior easier, in ways that would actually jeopardize the benefits of their deployment.

The problem at hand is to secure the operation of vehicular communication systems, that is, design protocols that mitigate attacks and thwart to the greatest possible extent deviations from the implemented protocols. Securing vehicular communications is a hard problem, with a broad range of challenges to be addressed.

Different aspects warrant distinct protocols, and thus per (type of) protocol specifications. Instead of such specifications, we provide next an outline of general security requirements. The listed below requirements do not serve as specification and are not necessarily relevant to all aspects of network operation and all applications. They are rather stand-alone requirements and can be viewed as building blocks towards more complex specifications.

Message Authentication and Integrity Messages must be protected from any alteration and the receiver of a message must corroborate the sender of the message. Integrity, however, does not necessarily imply identification of the sender of the message.

Message Non-Repudiation The sender of a message cannot deny having sent a message.

Entity Authentication The receiver is not only ensured that the sender generated a message but in addition has evidence of the *liveness* of the sender. A received unmodified message was generated within an interval $[t - \tau, t]$, with t the current time at the receiver and $\tau > 0$ a sufficiently small positive value.

Access Control Access to specific services provided by the infrastructure nodes, or other nodes, is determined locally by policies. As discussed further in Sec. VI, access to network and messages is mandated by default open to all nodes. This, however, does not preclude the need for fine-grained policies for all other purposes, as well as the assignment of distinct roles to different types of nodes. As part of access control, **authorization** establishes what each node is allowed to do in the network, e.g., which types of messages it can insert in the network, or more generally the protocols it is allowed to execute.

Message Confidentiality The content of a message is kept secret from those nodes that are not authorized to access it.

Privacy and Anonymity Vehicular communication systems should not disclose or allow inferences on the personal and private information of their users. This being a very general statement and a requirement within the broader area of information hiding, we state a narrower requirement within the vehicular network context: anonymity.[1]

We require **anonymity** for the actions (e.g., messages, transactions) of the vehicular network entities, which we denote as nodes, with respect to *a set of observers*. At *minimum*, any of the observers should *not* be able to learn if a node performed or will perform in the future a specific action, assuming that the node performs the action. Such a definition does not, however, guarantee that it is impossible for the observer to infer, with relatively high probability, the identity of the node that performs the action in question.

To prevent such inferences, stronger anonymity requirements would be necessary: nodes should be almost equally likely to have performed an action, or have *strong probabilistic anonymity*, with the probabilities, as far an observer is concerned, being equal for any node [23]. Or, without considering probabilities, require *full anonymity*: an action α performed by a node x could have been performed, as far as the observer is concerned, by any other node in the system.

The definition of anonymity depends on what is the set of the VC system entities. Or, in fact, whether entities are partitioned into a number of subsets, for administrative reasons. This implies that the anonymity requirement needs to be modified accordingly. For example, if two non-overlapping subsets A and B existed, a node $x \in A$ remains anonymous as long as x and any other node y also in A are equally likely to have performed action α. However, it may be trivial to infer that any node $z \in B$ did not and will not perform α.

Anonymity requirements could be refined further, for example, by considering the nature and capabilities of the observers. For example, observers could share information in different manners [30] in an attempt to either learn that a node x performed or is more likely than other nodes to have performed action α. Moreover, it is possible that anonymity is not a requirement with respect to special set of observers, due to a different system requirement we discuss below. Similarly, anonymity may not be a reasonable requirement for all entities of the vehicular communications system. A discussion of specific issues related to identity management and privacy enhancing technologies for VC can be found in [25].

Availability Protocols and services should remain operational even in the presence of faults, malicious or benign. This implies not only secure but also fault-tolerant designs, resilience to resource depletion attacks, as well as self-stable protocols, which resume their normal operation after the 'removal' of the faulty participants.

Liability Identification Users of vehicles are liable for their deliberate or accidental actions that disrupt the operation of other nodes, or the transportation system. The vehicular network should provide information that identifies or assists the attribution of liability.

This is a requirement that largely follows from the current

[1] Note that confidentiality is a different requirement.

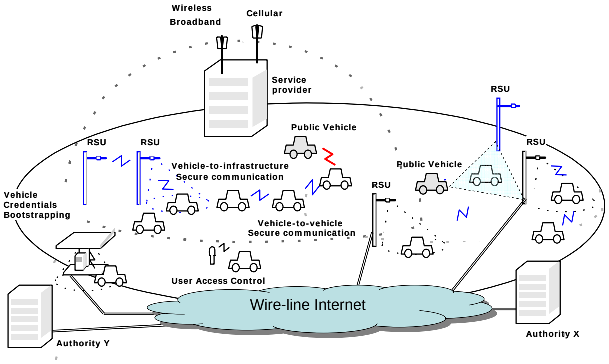

Fig. 1. Secure Vehicular Communications System - An Architectural View.

practice in transportation systems. However, liability identification implies that anonymity would need to be paired with the option to learn or essentially recover the node's identity if necessary. Specifying the types of observer (e.g., a public authority) that is vested with the power to do so depends on the actual scheme.

III. System Model

A vehicular communications system comprises a number of interacting entities that we classify broadly as: *(i) Users*, *(ii) Network nodes*, and *(iii) Authorities*. An illustration of the system, depicting some basic only communication and security operational aspects, is shown in Fig. 1.

Our focus is on the network operation and the communication of the computing devices, i.e., largely, the network nodes that we define more precisely below. Nevertheless, *users* (that is, individuals operating vehicles) are instrumental in determining the vehicle behavior and the overall transportation system operation, and thus warrant a distinction.

Network nodes are processes running on computing platforms capable of wireless communication; they are mounted on *vehicles* and *road-side units (RSUs)*. We denote the *RSUs* collectively as the *road-side infrastructure (RSI)*. The complexity of the nodes can vary from relatively powerful devices (e.g., on-board vehicle computers) to relatively simple ones (e.g., alert beacons on the road-side).

The *authorities* are public agencies or corporations with administrative powers in a specific field; for example, city or state transportation authorities. They are responsible for instantiating procedures, as those currently in place for vehicle registration and driver license issuance, as well as vehicular network entities that act on behalf of the authorities and provide services. For the rest of the discussion, we refer to authorities only as network entities, unless noted otherwise. A detailed discussion of the VC system operational assumptions follows.

A. Authorities

Authorities are trusted entities responsible for the issuance and management of *identities* and *credentials* for parties involved in the vehicular network operation. In general, authorities can be *multiple* and *distinct* in their roles and the subset of network parties in their jurisdiction.

We denote the set of system entities, S_X, registered with an authority X determined by geographical, administrative, or other criteria, as the *domain* of X. All parties in S_X trust X by default.

The presence of on-line authorities is not required, as connectivity and communication, especially over the wireless medium, with an authority may be intermittent. Nodes can in general establish two-way communication with the authorities, even though one-way communication, from an authority towards the nodes can be meaningful as well.

B. Vehicle Identification and Credentials

Each vehicle has a unique identity *V*, and a pair of *secret or private* and *public* cryptographic keys, k_V and K_V respectively. The binding of K_V to V and the binding of K_V to other data or *attributes* pertinent to V are achieved by an *identity certificate* or an *attribute certificate* respectively. We denote a certificate on K_V issued by an authority X as $Cert_X\{K_V, A_V\}$, with A_V being a possibly void attribute list.

The vehicle identity, V, denotes the *on-board central processing and communication module*. Other on-board sensing, actuating, and processing units are identifiable locally, with *V* having full control (access/read/write) over those resources. In other words, we abstract away the complexity of the on-board equipment, which could essentially be viewed as wired network of its own, as shown by the illustration of an in-car system by Daimler-Chrysler in Fig. 2. Thus, we consider a network node as:

i. a unique identity V
ii. a public/private key pair K_V, k_V
iii. a module implementing the networking and the overlying application protocols
iv. a module providing communication across a wireless network interface.

This abstraction, illustrated in Fig. 2 for a car, is applicable to vehicles and infrastructure nodes alike.

C. Infrastructure Identification and Credentials

Each infrastructure node has a unique identity, *I*, and k_I and K_I private and public keys. $Cert_Z\{K_I, A_I\}$ is a certificate issued by an authority Z for the infrastructure unit I with attribute list A_I.

A subset of the infrastructure nodes serves as a gateway to the authorities, or inversely, from the point of view of the authorities, a gateway to the mobile wireless part of the vehicular communications system. Infrastructure nodes, or a subset of those, can be considered as more trustworthy than other nodes, with respect to specific functionality or attributes. For example, infrastructure nodes can be assigned the role to

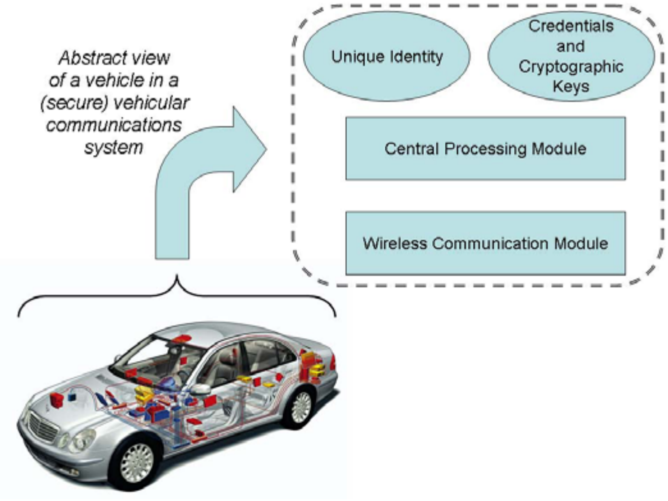

Fig. 2. Abstract view of VC system node.

transmit specific (safety or not) messages whose content is trusted as correct or given precedence over other messages. *RSUs* can be, for example, assumed to have absolute and relative *locations* that are in most cases fixed and thus often known or straightforward to infer.

1) Public Vehicles: Vehicles are distinguished in two categories, *public* and *private*. The former can include public-safety (highway assistance, fire-fighting) or police vehicles, aerial vehicles (e.g., police helicopters), or even public transportation vehicles (buses, trams). Public vehicles, similarly to infrastructure nodes, are considered more trustworthy, and they can be used to assist security related operations.

D. User Identification and Credentials

Each user of the vehicular communications system has a unique identity, *U*, and a pair of private and public cryptographic keys, k_U and K_U respectively. $Cert_Y\{K_U, A_U\}$ is a certificate issued by an authority Y for a user with an A_U attribute list. The user can be bound to its credentials and secrets through some token she/he uniquely knows or possesses (e.g., pass-phrase, biometric data).

E. User and Vehicle Association

The user can be the owner and/or the driver of the vehicle, or in general any passenger. The association of vehicles and users is in general *many-to-many*, however, at each point in time *only one* user can operate a vehicle. For the rest of the discussion, we make the simplifying assumption that the user is the individual that operates the vehicle, i.e., the driver. User access to the vehicle relies on the possession of a type of credential (e.g., physical key, PIN, biometric).

F. Trusted Components

Nodes are equipped with *trusted components (TCs)*, i.e., built-in hardware and firmware with two types of functionality: (i) *cryptographic* operations, and (ii) *storage*. The role of *TCs* is two-fold: to protect the vehicle's cryptographic material and their use, and to safeguard data usable for liability identification.

The *TCs enforce a policy* on the interaction with the on-board software, including the access and use of the securely stored keys, credentials, and secrets. Access (read or write) to any *information* stored in the *TCs* and modification of their *functionality* is possible only through the *interface* provided by the *TCs*. For example, the protected information cannot be exposed through the execution of any sequence of the commands provided by the interface. Similarly, the policies enforced by the *TCs* specify the authorized entities to modify information and functionality.

Cryptographic operations, with signature generations and verifications expected to be the more frequent ones, are performed without the *TC* revealing the cryptographic material to the potentially compromised or faulty computing unit. On-the-fly data and outcomes of computations are also stored, with those corresponding to a recent interval $[t_0, t]$ maintained if the *TC* is triggered by a specific event at time t, to provide protected *audit trails*.

The *TCs* can be *tamper-resistant*, in order to provide enhanced protection of the cryptographic material and other data. Tamper-resistance can also imply that keys, credentials, and other secret data are physically bound to the on-board platform. It is however possible to minimize or waive the requirement for tamper-resistance; for example, the points of the audit trail can be signed or encrypted data.

Assuming on-board *TCs* is in accord with the current state and developments in vehicle equipment, which already includes hardware components and firmware that regulate or record information on the vehicle operation and on its users' inputs. Examples are speed limiters, tachographs and *event data recorders* (*EDRs*) [12]. These may not be necessarily tamper-resistant but they are tamper-evident, and it is commonly accepted, among manufacturers (more so in the US) and legislators, that *TCs* will be routinely present.

IV. Communication model

In this section, we model the wireless communication in vehicular networks, whose connectivity and membership can change frequently, and so does the network area reachable by each node. We focus mainly on the data link layer, and then discuss other considerations. We model the communication by the following data-link (denoted by the subscript L) layer primitives and assumptions, for some radius R and time τ:

i. $Send_L(V, m)$: transmits message m to node V within radius R of the transmitting node.

ii. $Bcast_L(m)$: broadcasts message m to all nodes within radius R of the transmitting node

iii. $Receive_L(m)$: receives message m transmitted by a node within radius R of the receiver; m is processed at a receiver V if m was $Bcast_L(m)$ or $Send_L(V, m)$.

iv. A link (W, V) exists (or it is *up*) when two nodes W and V are able to communicate directly, i.e., W can receive transmissions from V and vice versa. We denote any two nodes connected by an *up* link, and thus capable of bidirectional communication, as neighbors.

v. Links are either *up* or *down*, and their state does not change faster than the transmission time of a single packet.

vi. The network connectivity, at a particular instance in time, can be modeled as the graph G the edges of which are all *up* links.

vii. Transmissions from W are received by all nodes V_i such that (W, V_i) is up during the entire duration of the packet transmission.

viii. Packets are delivered across an up link within a maximum link delay τ, or they are not delivered at all. In the latter case, the delivery failure is reported to the upper layer protocol. The data-link layer handles transient network failures, it retransmits, but it does not duplicate packets.[2]

Communication across the network is dependent on the availability of sufficient resources and, in particular, bandwidth. The shared medium implies that k nodes within R of each other contend and obtain a portion of the bandwidth, in principle, inversely proportional to k. We do not assume that the network provides any fairness and in general the available bandwidth can fluctuate, be unevenly distributed among neighbors, and links be congested. Failures can either be transient and thus masked by the data link layer, or they can persist and cause data or control traffic to be dropped, or prevent nodes from accessing the medium altogether. The latter case is equivalent to having all affected links at the *down*.

Our abstraction of R does not imply an idealized communication model; R is a nominal range within which direct wireless communication is possible. Yet, this can vary over time, and it depends both on the Physical Layer protocol and the Signal to Interference and Noise Ratio at the receiver. Moreover, different classes of nodes may operate with different values of R, with the ramification that this implies unidirectional communication.

It is also possible that nodes have more than one network interface. One option is to consider that multiple nodes 'run' on each platform, with a distinct identity and credentials for each interface. The other is to explicitly define such heterogenous interfaces (and thus links) and consider them as part of the node connectivity.

At the layers above the data link, different types of protocols can be implemented often according to the targeted applications. For example, a bound on the message reception delay can be meaningful and either *asynchronous* or *synchronous* communication may be possible according to the system requirements and the implemented network protocols. Next, we discuss three distinctive aspects of vehicular communications.

A. *Multi-domain and Highly Volatile Environment*

Any two or more nodes within range, or within multiple wireless links (hops), communicate independently of whether they are registered in different domains. This is so because nodes can be highly mobile and not bound to administrative and geographical boundaries. The connectivity and the set of nodes in the vicinity of a node can change radically over time, at different time scales. As a result, it is likely that nodes not registered with the same authority communicate and become associated.

B. *Frequent Broadcast Communication*

Unicast or multicast communication is clearly possible, as networking protocols may determine a set of intended receivers or simply a single receiver. However, a large portion of the vehicular network traffic is *broadcasted* at the network or application layers. Restrictions (e.g., based on the location of the sender and receivers) determine how a message propagates across the network. In fact, the sender of a message does not have, in general, prior knowledge of the receivers.

A second distinctive characteristic of VANET traffic is that messages are transmitted either *periodically* or *triggered* by in-vehicle or network events, or both. The transmission period is in general low (a fraction of a second; an indicative value of 0.3s), thus resulting in frequent transmissions.

C. *Time-sensitive Communication*

We do not dwell on the network and application protocols, as they clearly depend on the specific instantiations. The critical nature of vehicular communications implies nonetheless that a subset of the exchanged messages must be delivered in a timely manner in spite of the frequently changing network connectivity. Message delivery can be constrained by *deadlines*, with differing delay requirements for different types of messages.

V. Adversary Model

The behavior of an adversary can vary widely according to the implemented protocols and the capabilities of the adversary, whose incentive may be his own benefit or malice. In this section, we provide a general model for the adversary, or essentially a collection of adversary models, rather than describing protocol-specific actions of adversaries. Then, we discuss examples related to different types of network and system functionality and how the related adversarial behaviors fit the general model. We note, however, that we take into consideration the system assumptions in Sec. III, as it turns out that they can automatically restrict in some cases the adversary model.

Network entities can either comply with the implemented protocols, in which we denote them as *correct* or *benign*, or they may deviate from the protocol definition. In the latter case, we denote the nodes as *faulty* or *adversaries*. Faults may not be malicious; e.g., the communication module of a node may be discarding a message, or a bug causing a protocol to

[2] A distinction between the broadcast and the send (unicast) primitives can be made: τ_B and τ_S respectively, in general different, can be defined, depending on whether, for example, $Bcast_L()$ entails retransmissions or not.

set a packet field to an inappropriate value. Clearly, malicious behavior can result in a much larger superset of faults.

Before we elaborate on malicious behaviors, we first discuss types of *failures* that can be benign. *Crash* failures [6] are the simplest ones, with the node halting its operation. *Omission* failures [16] occur when a node fails to send a subset of its protocol-defined messages. *Timing* failures [7] occur when the node transmits messages *early* or *late*, or *never*. Crash, omission, and timing failures can also be attributed to the communication environment. Moreover, they are not protocol-aware, not requiring, for example, knowledge of the formats of the omitted messages.

Nevertheless, we are interested in the behavior of adversarial nodes. Viewing the adversary as the environment itself is a possible and broadly used approach [10] whose appropriateness for our setting is discussed in Sec. V-E. It is important to note that malicious behavior includes benign failures. An adversary can 'emulate' any of them by deviating from the protocol definition accordingly. Furthermore, malicious behavior can be protocol-aware: for any type of adversary we discuss here, protocol awareness is a given.

Looking at the simpler type, the crash failures, it becomes obvious that an adversary can withdraw itself from the execution of the protocol, i.e., cease to generate messages, at any point in time. Similarly, it can also elect to join again ('reboot') at a later point after the 'crash.' In this sense, an adversary can choose when it is part of the protocol and when not. It appears this is always true, independently of whether there may be consequences for not executing a protocol; e.g., not being granted access to data or services. Moreover, in a volatile communication environment, as defined above, deliberate *leaves* and *joins* from the protocol execution can be very hard to distinguish from actual connectivity changes.

Any wireless-enabled device that runs a rogue version of the vehicular communication protocol stack poses a threat. However, it is important to distinguish such devices from the vehicular network entities (nodes). In Sec. III, we defined nodes to be equipped with credentials and cryptographic keys; an adversary that does not possess such keys and credentials is by our definition excluded from being a node; we term such an adversary as an *external adversary*. In contrast, *internal* adversaries are, as per our system model, network nodes, that is, entities equipped with credentials that entitle them to participate in the execution of protocols. It is implied that crash, omission, and timing failures, and thus the corresponding malicious behavior, are relevant to legitimate participants, i.e., internal adversaries.

Next, we consider a *passive eavesdropper*, an adversary that can only intercept messages to extract or infer information from those messages. More generally, a *passive adversary* learns information about network nodes, but cannot affect or change their behavior. In contrast, an *active adversary* controls or affects the operation of network entities (nodes), in addition to its ability to learn information about system entities.

Passive adversaries can clearly be external. Active adversaries can be external as well, in which case they can affect the operation of network nodes even though they do not control their behavior (e.g., protocols they implement). An external active adversary cannot generate and inject message as a legitimate system entity, yet it can act so that it appears to be part of the communication environment.

This is possible as an external active adversary can *replay* messages, that is, transmit one or more times messages that were previously transmitted by other nodes and the adversary received. For this, it suffices to implement the communication part of the protocol stack.

A. Localized and Selective Denial of Communication

An (external) adversary can deliberately generate interfering transmissions and prevent (*jam*) communication within the reception range of those transmissions. Similarly to the communication model, we assume an upper bound R_{jam}^{max} on the range affected by a jammer. Essentially, this can take the form of network resource depletion, with an adversary denying communication to all receivers within its range.

It is also possible that such an active adversary jams, i.e., erases one or more messages, selectively. This is possible if the targeted message(s) sender and receiver are both within range of the adversary. The denial of communication can be achieved through disruption at different layers of the protocol stack (e.g., medium access control).

B. Internal Active Adversaries

In the rest of this section, we consider internal active adversaries, whose behavior is a superset of that of external (active) adversaries. We emphasize that the possession of credentials and cryptographic material does not guarantee the correct operation of the nodes; for example, nodes can be compromised or infected. We also note that an internal adversary can be passive; this would be a significantly weaker adversary, meaningful only if it sought to avoid detection, and we do not discuss it further.

As mentioned above, an adversary can omit, delay, or transmit early messages it generates or relays as per the protocol definition. A more general type of adversary *modifies* in-transit messages it sends. An arbitrary alteration or corruption of messages is also a modification; yet, simple checks at the input to identify messages that deviate from the protocol-defined format can render a large class of modifications irrelevant.

An adversary can *forge*, that is synthesize in a manner non-compliant to the protocols and system operation, and *inject* messages. Such adversaries include those that modify in-transit messages. The basic difference of adversaries that inject (forged) messages from all those aforementioned is that they can initiate a protocol run.

The choice of which message to forge, replay, modify, omit, or delay, may be independent of the message content or *message-oblivious* [8]. Or it may be *message dependent*, i.e., depend on the message content or messages the adversary previously received. Adversaries can *recollect* past messages they receive, as well as run algorithms to maintain a summary of them so that essentially they recollect all received messages;

the bounded capabilities of adversaries are discussed further below.

Multiple adversarial nodes can be present in the network. Each adversary can behave in any of the above discussed models. In general, adversaries can follow the stronger model that includes all above types of behaviors. Adversaries can be *independent*, with each selecting its actions (attacks) independently of those of other adversaries. Or, adversaries can *collude*, i.e., coordinate their actions by exchanging information, including their view of the protocol execution, and their local states. From the assumption of the *Trusted Components*, however, it follows that colluding adversaries are prevented from exchanging cryptographic material and credentials.

The presence of multiple adversaries, especially in the case of colluding ones, implies that the set of adversaries could be viewed as a ”collective” single entity. This would be in agreement with the view of the adversary as an entity that compromises one or more network entities.

The set of adversarial nodes may be *fixed*, in this case corresponding to the definition of an *non-adaptive adversary*. Or, the set of adversarial nodes can *change* over time, with the adversary selecting which entities (thus protocol participants) to compromise; this would be an *adaptive adversary* [5]. In the case of an adaptive adversary, it is possible to have nodes that remain compromised throughout the protocol execution, once they are subverted by the adversary. Or, the adversary may attack and control one node for a period of time, before moving to another node. This is a *mobile adversary* [24]; more general, the set of compromised nodes can change over time.

We assume that adversaries are *computationally limited*, i.e., have polynomial resources and computational power. This implies that hard problems (e.g., discrete logarithm, factorization) cannot be solved, and thus correctly implemented cryptographic primitives are considered secure. Moreover, the knowledge of an adversary is algorithmic and also limited (no logical omniscience) [17]. Note, however, that a computationally bounded adversary can still attempt, and succeed with a very low probability, guessing attacks [22]. Finally, a bounded adversary implies that its memory cannot be infinite. However, recollection of *all* messages of a protocol run is in practice possible if the adversary can summarize the history of the protocol messages.

C. Bounded Adversarial Presence

Overall, at most a (small) fraction $t << 1$ of the network nodes are adversaries. Such a bound as well as a bound on the transmission range of each adversary can yield a bound on the fraction of the network area within which communications can be intercepted or affected by adversaries. Clearly, this will depend on the distribution of the correct nodes locations (coordinates) and the placement of the adversarial nodes.

In the cases of adaptive and mobile adversaries, the bound on the fraction of adversarial nodes can be defined with respect to a *time period* (perhaps, the period of the protocol execution, if defined) or any *time window*. Such a bound would determine the rate at which an adversary increases its penetration in the network.

The bound on the presence of adversaries can be further refined by having distinct thresholds for different node types. For example, no more than t_V per cent of private vehicles and no more than t_P per cent of public vehicles can be compromised by an adaptive adversary. Or, no more than t_I road-side infrastructure nodes can be compromised by a mobile adversary at any point in time. Such a distinction can reflect the relative hardness of compromising nodes of different types; e.g., $t_P < t_V$.

The above bounds do not preclude that a few adversarial or faulty nodes surround a correct node at some point in time. However, the scale, volatility, and diversity of vehicular communication systems clue that a correct node is more likely to encounter a small number of adversaries. This, accompanied by the conjecture that adversaries are more likely to be independent than colluding, was denoted as *adversarial parsimony* and was the basis of heuristics in [15] essentially following the principle of ’Occam’s Razor.’

D. Input-controlling Adversary

An adversary may be unable to deviate from the implemented protocols, but it may be capable to alter the local inputs into the protocol. In the context of vehicular networks, this can be the case if the sensory outputs provided to the on-board processing unit are tampered with. The complexity of such an adversary can be in general lower than that of an adversary that modifies and deviates from the protocol operation. Especially if the trusted processing component protects the protocol functionality.

It is not straightforward to classify the input-controlling adversary as an external or internal adversary. On the one hand, no access to credentials and cryptographic material is necessary. On the other hand, messages due to the input-controlling adversary are generated and transmitted as originating from a legitimate system participant.

E. Discussion of other Adversary Models

The strongest model for active internal adversaries is essentially the model of *Byzantine* adversaries [20], with the addition of the capability to erase messages of other nodes. Byzantine faults imply that the adversaries are legitimate participants of the protocol(s).

This model is a powerful one, as an adversary can generate and send arbitrary messages at any point in time and it encompasses, yet not expressly, collusion. Overall, Byzantine adversaries can implement complex algorithms without, however, being able to generate any form of authenticator without the corresponding secrets or keys. The assumption on adversarial parsimony does not preclude Byzantine behavior. It only implies that the likelihood of interacting with an adversary is inversely proportional to the complexity of the adversary algorithm (capabilities).

Compared to such adversaries that can arbitrarily deviate from the protocol definition, the *input-controlling adversary* is weaker. However, it is interesting to compare it with arbitrary malicious protocol participants which are assumed (for example, in the case of secure multi-party computation protocols) to fully control their local input. In contrast, the input controlling adversary cannot create any arbitrary behavior, but only induce the node's protocol behavior by providing input values accordingly. For example, fabricated measurements that warrant an 'alarm' can trigger the transmission of corresponding messages.

The capability to intercept and inject any message, without necessarily being a part of the system, has been part of another widely used model, the *Dolev-Yao (DY) adversary* [10]. A *DY* adversary is an 'active eavesdropper,' able to obtain *any* message transmitted across the network, i.e., receive any message of a protocol execution ('conversation') initiated by any other network participant. Moreover, the *DY* can delete any such message that traverses the network and can initiate a 'conversation' with any network participant. However, the public key cryptosystem in use is assumed 'perfect,' i.e., the *DY* adversary is incapable of learning anything about a message or its digest unless it possesses the necessary key. It is also implied that keys are under the control of the principals and used only for a specific protocol under consideration.

The *DY* is also a very powerful adversary model, commonly assumed as the adversary model in the Internet community. *DY* exceeds the capabilities of the adversaries we discussed above, yet does not capture several important aspects relevant to our system.

In what follows, we discuss only the relevance of the *DY* adversary and not the *DY* model of cryptographic protocols; this method of modeling and analysis was extended and discussed for example in [13] towards multi-party protocols, or [2] to adapt the model for web services. We do not dwell on the cryptographic limitations of the *DY*, which does not consider the case of guessing attacks.[3] We have already assumed that adversaries have bounded resources and with very low probability may forge a signature or invert a one-way function; these assumptions are not dependent on the employed cryptographic primitives.

At first, *DY* in our context implies an *omnipresent* set of adversarial nodes, covering the entire area of the vehicular inter-network in order to intercept all wireless hop-wise communications. This contradicts our assumption of bounded adversarial presence. More important, it would be an *unrealistic* assumption, as the aggregate resources of the adversaries are still bounded, yet they would grow at least at the rate the network itself grows.

A second point of caution for the *DY* stems from what could be loosely defined as the 'awareness' of the adversary. *DY* does not consider the inferences an adversary can make from intercepted messages. In the context of authentication protocols, [27] gives a characteristic example: the protocol considered is such that the holder of an n-bit key transmits one bit of the key in each message, yet it is not possible to argue whether the *DY* obtained the key even though it would clearly capture the entire key after n messages. Such lack of awareness limits, in practice, the effectiveness of the adversary's actions (attacks), even though conceptually the adversary could inject any arbitrary message. As an example, consider an adversary that 'interprets' the received safety-related messages reporting vehicle sensor readings, and injects accordingly a false-valued message, versus one that could elect and forge the same one among all possible forged message values.

VI. Design Principles

Many proposals to address security for vehicular communications are expected to be devised. Nevertheless, security mechanisms and protocols to safeguard the system operation and thwart adversarial behavior will in general differ in functionality. Based on our investigation on operational assumptions, the system, communication, and adversary models, as well as experience from other related networking paradigms and the to-date development of the vehicular communications technology, we propose next a set of twelve design principles for future security solutions.

A. Default Network Access

Messages, especially broadcasted (e.g., safety, driver assistance) are by default accessible to all nodes that can receive them. Similarly, nodes are by default assumed to assist multi-hop communication. Furthermore, possession of keys and valid credentials is the basic prerequisite for nodes to transmit messages.

B. Locality and Timeliness as Privileges

As vehicular networks and the supported applications are context-aware, only the vicinity of a node to a location or an action to a point in time may enable specific action. Examples are the generation or validation of a specific message or a credential, the request and access to a service, or the participation to a distributed protocol execution.

C. Visibility of Events

An attestation of an event by an individual node requires that the event be visible to the attesting node; either the node is the sole responsible for generating the event (e.g., alarm), or it had the locality and timeliness privilege (e.g., reception of the message within δ seconds from its generation) to provide the attestation. More generally, events that trigger joint computations or actions by multiple nodes should have been visible to all nodes participating in the distributed protocol.

D. Mandated (non-circumventable) Mediation

All actions that change the security state of the network (e.g., assignment of identities or distribution of cryptographic keys and credentials), are mediated by a network authority. Network authority actions cannot be circumvented by any network node. For example, no coalition of nodes can substitute an authority and issue new or revoke valid credentials.

[3] From a different point of view, the independence of *DY* from the cryptosystem is positive.

E. Accountability

All messages or protocol executions that affect the network operation, or at least a critical subset of the operation, and in particular the participation of other nodes in the network, are *auditable* by the authorities. For example, alarm messages that notify of system malfunctions must be auditable.

F. Vehicle Autonomy

Node actions and operations that do not require mediation are autonomous with respect to those of other nodes. For example, all nodes are able to reject messages from other nodes, or utilize solely their local input for any computation. We note that autonomy, however, does not mean freedom in participating in the protocol execution, e.g., avoid relaying messages.

G. Separation of Privilege

Reliance on multiple authorities, as well as the separation of the roles of authorities and infrastructure, and thus the distribution of trust, can provide increased security, privacy, and fault-tolerance.

H. Non-frameability

Non-frameability implies that a trusted entity (node) cannot perform actions or more generally prove that a node x performed the action, if x never did so. This principle is based on our operational assumption of authorities that nonetheless should not be all-powerful. For example, use of a cryptosystem that allows the authority to sign on-behalf of the registered entity (vehicle) would not satisfy this principle.

I. Staged Response to Faulty Behavior

This principle calls for a system design with a multi-level, escalating response to faults. More specifically, a low assurance detection can be followed by a warning, then self-constrained participation, a probation period, and local containment following a report, and finally eviction from the system. At all stages, reinstatement should be possible.

The cause behind such a principle is the difficulty in distinguishing among benign and malicious faults due to the network volatility, and the frequent non-critical nature of the majority of faults. The bounded adversary presence is related, as it renders the staged response a reasonable approach. Which can also be assisted by the inherent redundancies in the protocol mechanisms. Moreover, actions at different stages can be in accord with the principle of vehicle autonomy. Finally, eviction clearly assists towards bounded adversarial presence.

J. Reconfigurability

Reconfigurability is a principle that applies to the on-board software and firmware (e.g., automatic download of patches), as well as the policies that describe what services each node provide to its peers. More generally, reconfigurability of services provided by the infrastructure nodes (registration/de-registration), as well as flexible service discovery (and delivery). Open interfaces to the network and security services.

Beyond its obvious practical aspects, reconfigurability can be viewed as the means to ensure the bounded adversarial presence. In that sense, it is related to the response to faulty behavior, and be viewed as an additional means to ensure bounded adversarial presence. For example, applying software patches and updating virus definition files can prevent the exploit of software vulnerabilities and the spread of malicious software across the VC system.

K. Privacy Conservation

Vehicular communications should not become a weak link in terms of privacy, providing users *at least* with the same level of privacy that is currently afforded without vehicular networks. The privacy of a driver should be protected against private citizens and law enforcement agencies. Conservation is meaningful in the latter case as well, whereas privacy should be conditional to specific scenarios (liability).

L. Usability

Usability calls for ease of the user to access and utilize the vehicular communication system, as well as to utilize the information it provides. Psychological acceptability is one important aspect. Simplicity and reliability of management operations (e.g., maintenance, or refreshing of credentials) is also crucial; consider, for example, the clearly unwanted situation that a vehicle computer would not allow its engine to start because a necessary 'fresh' credential cannot be obtained.

VII. Discussion and Conclusions

In this paper, we have proposed a system model, making a minimal set of operational assumptions. On the one hand, these are sufficiently restrictive, to take into consideration salient features of vehicular communication systems. On the other hand, they are generic in that they allow for a wide range of architectures and refinements. Furthermore, we did not dwell on specific applications, networking protocols, communication technologies, or cryptographic primitives.

In terms of communications, we provided a general model that captures the distinctive characteristic of VC, the V2V communication, but not only (e.g., V2I is also covered). The model can be satisfied by a range of protocols, including the the emerging defacto standard, DSRC (Dedicated Short Range Communications), which is on the IEEE 802.11 technology and proceeds towards standardization under the name of IEEE 802.11p [11]. Our model can be extended to encompass other technologies, such as cellular telephony data (an integration with 802.11 is proposed by [9]) and wireless broadband [31], as well as directional antennas.

Regarding the adversary model, we outline a family of adversaries again without considering specific applications and protocols. Through a careful investigation of the literature, we find that we cover previously described attacks [1], [32], [19], [3], [15], [26], [14], [28], [21], [18], [29]. Another advantage of providing a family of adversary models is that adversarial behaviors can be composed by any meaningful combination of those models. This is important as one cannot anticipate in detail any attack against any (not yet defined) protocol. Finally, one can classify adversaries according to their sophistication and thus likelihood to be encountered. We have not provided such a classification here, yet, our assumption on bounded adversarial presence or the identification of input-controlling adversaries, clearly point towards what could be called the more 'realistic' or 'expected' models. A more detailed treatment of this topic, which can also become more protocol- and application- specific, is part of our on-going work.

The security requirements and design principles we have presented here are also independent of the system functionality. The outlined requirements can be used as building blocks, but the presented list could have been longer only if we considered particular protocols and attempted to provide specifications. The list of design principles could also grow, in a seemingly straightforward manner, to include principles such as open design or graceful degradation of performance in the presence of increasing-strength attacks.

Our investigation in this paper, focused on VC systems, is largely relevant to other wireless and mobile communication systems, such as mobile ad hoc networks, sensor networks, and infrastructure-based wireless networks [4]. Our intention however is also to identify the elements that distinguish VC systems from other networking paradigms. Our focus is on VC, which emerge as a promising technology that draws world-wide support and has the potential for large-scale deployment. We believe that this paper can be the basis for future designers of security solutions for vehicular communication systems.

ACKNOWLEDGEMENTS

The authors would like thank Stefano Cosenza, Albert Held, Frank Kargl, Rainer Kroh, Tim Leinmueller, and Hans-J Reumerman for their insightful comments on an early draft of this paper, Albert Held for the graphic used in Fig. 2, as well as the LCA1 and SeVeCom members for many useful discussions.

REFERENCES

[1] A. Aijaz, B. Bochow, F. Dtzer, A. Festag, M. Gerlach, R. Kroh and T. Leinmuller, "Attacks on Inter-Vehicle Communication Systems - An Analysis" *in Proceedings of the 3rd International Workshop on Intelligent Transportation (WIT 2006)*, March 2006

[2] M. Backes and T. Gross "Tailoring the Dolev-Yao abstraction to web services realities - a comprehensive wish list" *Proceedings of SWS'05*, Nov. 2005

[3] M. Blum and A. Eskandarian, "The Threat of Intelligent Collisions," *IT Professional,* January 2004

[4] L. Buttyan and J-P. Hubaux, Security and Cooperation in Wireless Networks, Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing Cambridge Press (to appear), On-line: http://secowinet.epfl.ch/

[5] R. Canetti, U. Feige, O. Goldreich, and M. Andnaor "Adaptively secure multi-party computation," *Proceedings of the 28th ACM Symposium on the Theorg of Computing (STOC)*, 1996

[6] F. Cristian, "Understanding fault-tolerant distributed systems," *Communications of the ACM,* Volume 34, Issue 2, Feb. 1991

[7] F. Cristian, H. Aghili, R. Strong, and D. Dolev, "Atomic broadcast: From simple message diffusion to Byzantine agreement," *Proceedings of the Fifteenth International Symposium on Fault-Tolerant Computing*, p. 200-206, June 1985. Also, *IBM Research Laboratory Technical Report RJ5244*, Apr. 1989

[8] B. Chor, M. Merritt, D. B. Shmoys, "Simple Constant-Time Consensus Protocols in Realistic Failure Models," *Journal of the ACM*, Volume 36, Issue 3, July 1989

[9] "CVIS: Cooperative Vehicle Infrastructure Systems," URL: *http://www.ertico.com/en/activities/efficiency__environment/cvis.htm*

[10] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory,* Volume 29, Issue 3, March 1983

[11] "DSRC: Designated Short Range Communications," URL: *http://grouper.ieee.org/groups/scc32/dsrc/index.html*

[12] "Event Data Recorder Applications for Highway and Traffic Safety," *US National Highway Traffic Safety Administration - Research and Development* URL: http://www-nrd.nhtsa.dot.gov/edr-site/, 2006

[13] S. Even and O. Goldreich "On the security of multiparty ping-pong protocols," *Proceedings of the 24th FOCS,* 1983, updated version, TR-285, Technion, Haifa, 1996.

[14] M. Gerlach, "VaneSe - An approach to VANET security," *in Proceedings of the V2VCOM*, 2005

[15] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," *in Proceedings of the 2004 Workshop on Vehicular Ad hoc Networks (VANET)*, 2004

[16] V. Hadzilacos and J. Y. Halpern, "Message-optimal protocols for Byzantine agreement," *Mathematical Systems Theory*, 26, pp. 41-102, 1993

[17] J. Y. Halpern, "Reasoning about knowledge: a survey," *Handbook of Logic in Artificial Intelligence and Logic Programming,* Vol. 4, D. Gabbay, C. J. Hogger, and J. A. Robinson, eds., Oxford University Press, 1995

[18] "IEEE P1609.2/D2 - Draft Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages," November, 2005

[19] M. Jakobsson, X. Wang, S. Wetzel, "Stealth Attacks on Ad-hoc Wireless Networks," *in Proceedings of the Fall IEEE Vehicular Technology Conference (VTC-Fall 2004),* 2004

[20] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *Journal of the ACM*, Volume 27, Issue 2, Apr. 1980

[21] M. Li, R. Poovendran, K. Sampigethaya, and L. Huang, "CARAVAN: Providing Location Privacy for VANET," *in Proceedings of the Embedded Security in Cars (ESCAR) Workshop,* Cologne, Germany, November 2005

[22] G. Lowe, "Analysing protocols subject to guessing attacks." *Proceedings of the Workshop on Issues in the Theory of Security (WITS02)*, 2002

[23] K. O'Neill and J. Y. Halpern, "Anonymity and information hiding in multiagent systems," *Proceedings of the 16th IEEE Computer Security Foundations Workshop*, 2003

[24] R. Ostrovsky and M. Yung, "How to withstand mobile virus attacks," *Proceedings of the Tenth ACM Annual Symposium on Principles of Distributed Computing (PODC91)*, Montreal, 1991

[25] P. Papadimitratos, A. Kung, J-P. Hubaux, and F. Kargl, "Privacy and Identity Management for Vehicular Communication Systems: a Position Paper" *Workshop on Standards for Privacy in User-Centric Identity Management*, Zurich, Switzerland, July 2006

[26] B. Parno and A. Perrig, "Challenges in Securing Vehicular Networks," *in Proceedings of the Workshop on Hot Topics in Networks (HotNets-IV)*, 2005

[27] R. Pucella and J. Y. Halpern, "Modeling adversaries in a logic for security protocol analysis," *Proceedings of Formal Aspects of Security,* 2003

[28] M. Raya and J-P. Hubaux, "The security of vehicular ad hoc networks," *in Proceedings of the Workshop on Security in Ad hoc and Sensor Networks (SASN)*, 2005

[29] M. Raya, P. Papadimitratos, and J-P. Hubaux, "Securing Vehicular Networks," *IEEE Wireless Communications*, Volume 13, Issue 5, October 2006 (to appear)

[30] P. F. Syverson, and S. G. Stubblebine, "Group principals and the formalization of anonymity," *Proceedings of the World Congress on Formal Methods,* 1999

[31] "The IEEE 802.16 Working Group on Broadband Wireless Access Standards," URL: http://www.ieee802.org/16/

[32] M. El Zarki, S. Mehrotra, G. Tsudik, and N, Venkatasubramanian, "Security Issues in a Future Vehicular Network", *in Proceedings of European Wireless*, 2002