5

Security and Privacy Mechanisms for Vehicular Networks

Panos Papadimitratos

Ecole Polytechnique Fédérale de Lausanne

The recent increase in interest and developments in the area of Vehicular Communication (VC) systems indicate that the technology could become broadly available in the near future. At the same time, authorities, industry, and researchers in academia agree that security and privacy enhancing mechanisms are a prerequisite for the acceptance and deployment of VC technology. A number of concerted efforts have been undertaken, with various projects providing significant results. As these approaches have common elements, this chapter surveys the state of the art in security and privacy enhancing methods for VC systems. A considerable distance has already been covered, but there is some uncertainty on how exactly VC systems are to be instantiated. Even though there is convergence in terms of securing communication, there are still challenges regarding other system aspects. A discussion on steps towards deployment and on the future landscape concludes the chapter.

5.1 Introduction

Intelligent Transportation System (ITS) and related technologies have been deployed in recent years for toll collection, fleet logistics and management, anti-theft protection, pay-as-you-go insurance, traffic information, and active road-side signs. These systems, relying on various communication technologies, continue to evolve and proliferate. More recently, a new trend has emerged: on-board computing units (On-Board Units, or OBUs) and short-range high bit-rate radios (in addition to cellular network transceivers) are integrated into vehicles, and dedicated (likely sparse) road-side infrastructure is expected to be deployed. Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, or, in general, Vehicular

Vehicular Networking: Automotive Applications and Beyond Edited by Marc Emmelmann, Bernd Bochow and C. Christopher Kellum
© 2010 John Wiley & Sons Ltd. ISBN: 978-0-470-74154-2

Communication (VC) would enhance transportation safety and efficiency and would support various other applications. V2V communication would enable real-time safety applications, extending the driver's horizon, while both V2V and V2I communication would enhance, for example, the distribution of environmental and traffic conditions information and in-vehicle entertainment.

The unique features of VC are a double-edged sword: a rich set of tools would be offered to drivers and authorities but a formidable set of abuses and attacks would become possible if the appropriate safeguards were not in place. An attacker could 'contaminate' large portions of the vehicular network with false information, announcing, for example, non-existent dangerous or congested road conditions, and thus mislead drivers and cause traffic jams. Alternatively, drivers could purchase software or hardware VC system 'hacks', just as they now often purchase police radar detectors or modify their cars for additional horsepower. Such VC system modifications could, for example, allow private vehicles to transmit messages as if they were an emergency vehicle (e.g. ambulance, police patrol, or road maintenance vehicle), or they could have unsuspecting drivers notified by their OBUs to slow down and yield, and in this way offer fast movement for some vehicles even in traffic jams. From a different point of view, receivers deployed in a city center, at highway exits, or even in a celebrity's neighborhood could record transmissions from passing vehicles to be used later in tracing their location and inferring private information about their passengers.

Privacy Enhancing Technologies (PET) are needed, especially because attacks are relatively easy to mount. To begin with, VC relies on a variant of the widely adopted IEEE 802.11 wireless communication technology. Besides this, attackers could use any low-cost computing platform, such as palmtop or laptop computers, or wireless local area network Access Points (APs); for example, a wireless network operator, licensed to provide services unrelated to VC systems, could 'tune' its APs to intercept VC traffic. Finally, VC equipment can be left unattended for long periods, increasing the likelihood of physical compromise. Overall, without security, VC systems could make antisocial and criminal behavior easier, in ways that would actually jeopardize the benefits of their deployment.

The awareness of the need to secure VC has spurred a number of projects to design VC security architectures: the Secure Vehicular Communication (SeVeCom) project and the IEEE 1609.2 working group are two prominent efforts that seek to secure communication and protect private user information. The envisioned systems rely on multiple Certification Authorities (CAs), with each CA managing identities and credentials for nodes (vehicles and Road-Side Units, or RSUs) registered within its *region* (e.g. national territory, district, or county). Each node is uniquely identified and holds one or more private–public key pairs and certificates, thus digitally signing messages it transmits.

In this chapter we survey security and PET solutions for vehicular networks. With significant commonalities among approaches to secure communication, we ponder whether these aspects of the overall problem of securing VC systems are addressed. More generally, are there research challenges to be addressed, or is it rather clear which VC security architecture would be deployed? We reflect on these questions, and discuss alternative viewpoints and non-technical factors that are likely to influence the deployment of secure VC systems.

In the rest of the chapter we first discuss threats and security requirements, followed by an overview of a secure VC and a set of basic system assumptions in Section 5.4. We present secure communication schemes that also enhance privacy in Section 5.5.1. Then we discuss approaches for data-centric security, notably secure localization with the help of Global Navigation Satellite Systems (GNSS) (Section 5.7.1), and data trustworthiness (Section 5.7.2). After that we briefly discuss considerations on design choices and challenges to deployment, before giving our conclusion.

5.2 Threats

VC systems comprise network nodes, or, in other words, wireless-enabled computing platforms mounted on vehicles and RSUs. Their complexity varies from relatively powerful devices (e.g. vehicle OBUs or servers run by authorities) to relatively simple ones (e.g. alert beacons on the road-side). These VC entities can be *correct* or *benign* (i.e. comply with the implemented protocols) or they can be *faulty* or *adversarial* (i.e. deviate from the protocol definition).

Faults might not be malicious; for example, the communication module of a node may discard or delay messages or set packet fields to inappropriate values. We do not consider here benign faults, such as communication errors, or message delay or loss, which can occur either under normal operational conditions or due to equipment failure. Malicious behavior can result in a much larger set of faults. Papadimitratos et al. (2006a) provide a detailed discussion of faults and adversary models, aspects germane to VC systems, and models used in other types of distributed systems.

The behavior of adversarial nodes can vary widely, according to the implemented protocols and the capabilities of the adversary. The incentive of the adversary may be its own benefit, or it may be malice. Active adversaries can meaningfully *modify* in-transit messages, beyond what the protocol definitions allow or require them to modify. More generally, they can *forge* and *inject* messages, based on their own prior observations (messages they received) and the protocol they are attempting to compromise. An active adversary may also *jam* communications (i.e. interfere deliberately and prevent other devices within its range from communicating). Alternatively, it can *replay* messages it has received that were previously transmitted by other system entities. In contrast, *passive adversaries* only gather information about system entities and cannot affect or change their behavior.

The adversary can be *external* but still be able to influence the protocol execution, by jamming communications and replaying the messages of correct nodes. Alternatively it can be *internal*, with cryptographic keys and credentials to participate in the execution of the protocol(s). Even though the VC implementations would be proprietary, standards, needed for interoperability, would provide extensive information on the VC protocol stack. Attackers could in principle clone its functionality, build their own rogue protocols, and modify the functionality of VC system nodes. If they obtain compromised cryptographic keys, by physically extracting them from a node for example, then they can act as internal adversaries. In fact, a node holding multiple such keys can appear as multiple nodes.

More generally, many adversarial nodes can be present. Often, they can act *individually*, but they might also act in *collusion*, coordinating their actions. It is, however, likely that colluding adversaries are unwilling to share their private keys and allow other nodes to fully impersonate them (and obtain, for example, their access rights). Over time, the number of adversaries can change, depending on the type of compromise and the defensive reaction of the system. It is reasonable to expect that at any point in time a small fraction of the network nodes are adversaries. At any time and location, only a few adversaries are likely to be physically present. This does not preclude a group of adversarial nodes surrounding an honest one, which should be a rare situation.

A rather peculiar type of adversary is relevant to VC systems: an *input-controlling adversary* that alters (sensory) inputs to VC protocols, rather than compromising the protocols. Such an adversary is weaker than an arbitrary internal adversary, because it cannot induce arbitrary behavior. But it would often be much easier to affect inputs, or compromise sensors or sensor-to-OBU connections, than to compromise the OBU itself. Such an adversary is weaker than an internal one: controlling inputs alone cannot induce arbitrary behavior if self-diagnostics and other controls are available and out of reach of the adversary.

5.3 Security Requirements

In general, we seek to secure the operation of VC systems, or, in other words, to design protocols that mitigate attacks and thwart deviations from the implemented protocols to the greatest possible extent. Each protocol has its own specifications, but instead of requirements per protocol and application, stand-alone security requirements have been considered, largely independently of specific applications and protocols.

Message authentication and integrity mechanisms protect messages from alteration and allow receivers to corroborate the node that created the message. If necessary, *entity authentication* can provide evidence of the sender *liveness* (i.e. the fact that the sender generated a message recently). To prevent a sender from denying having sent a message, *non-repudiation* is needed. Furthermore, *access control* and *authorization* can determine what each node is allowed to do in the network, in terms of the implemented system functionality. *Confidentiality* can keep message content secret from unauthorized nodes.

Privacy and anonymity are required, at least at the level of protection achieved before the advent of VC systems. In general, VC systems should not allow disclosure of private user information. In particular, the identity of a vehicle performing a VC-specific action (e.g. transmitting a message) should be concealed. Anonymity, with respect to an observer, depends on the set of involved vehicles: an observer cannot determine, among all vehicles in a set, which vehicle performed an action. Moreover, any two actions by the same vehicle cannot be linked. But under specific circumstances an observer could consider a vehicle more likely to perform an action.

Rather than seeking *strong anonymity*, along with authentication and other security properties, less stringent requirements are considered. Cryptographically protected messages should not allow for the identification of their sender, and two or more messages generated by the same vehicle should be *difficult to link* to each other. More precisely, messages produced by a node over a protocol-selectable period of time τ can be linked, but messages m_1, m_2 generated at times t_1, t_2 such that $t_2 > t_1 + \tau$ cannot. The shorter τ is, the fewer the linkable messages are, and the harder it becomes to trace a node.

Beyond security and anonymity, *availability* is also sought, so that VC systems remain operational even in the presence of faults, and resume normal operations after the removal of the faulty nodes. Another significant dimension is that of *non-cryptographic security*, including the determination of data *correctness* or *consistency*. Traditionally, if the sender of a message is trusted, then the content of the message is trusted as well. This notion is valid for long-lived, static trust relationships. But in VC systems there are often no grounds for similar approaches. It is thus necessary to assess the *trustworthiness of data* per se, as obtained by other nodes in the VC system.

These general requirements can be mapped to specific VC-enabled applications. Ideally, one could argue that all requirements must be satisfied for all applications (Papadimitratos et al. 2006a). But the relative importance of security requirements can differ across applications. General application characteristics and security requirements were assessed for a large number of VC applications; for example, for a roadwork zone warning application, it may be relatively less important to rigidly determine the recency of its messages than for a collision avoidance application (Kargl et al. 2006; Papadimitratos et al. 2008a). Of course, for both applications it is critical to ensure that no message content can be fabricated by an attacker. Privacy protection is not required for infrastructure- or public vehicle-sent messages (e.g. the work zone and emergency vehicle warnings).

5.4 Secure Vehicular Communication Architecture Basic Elements

Efforts are being made in academia and industry with the aim of providing adequate security solutions: for example, the IEEE 1609.2 trial-use standard (IEEE 1609.2 2006), the Network On Wheels (NOW) project (NoW 2007), and the SeVeCom project (Kargl et al. 2008a; Papadimitratos et al. 2008a; SeVeCom 2009). Essentially, security architectures first seek to address two fundamental issues: (a) *identity, credential, and key management*, and (b) *secure and privacy enhancing communication*. The focus is primarily on securing the wireless part of the VC system and enhancing the privacy of its users, in order to satisfy the requirements outlined in the previous section. Additional aspects, such as the *in-car system protection* and the *data trustworthiness*, have received relatively less attention. In this section we present an overview of the basic elements of secure VC systems. Then, in the sections that follow, we give a more detailed presentation of secure communication, revocation, and mechanisms for data trustworthiness.

5.4.1 Authorities

Authorities are trusted entities responsible for the issuance and management of *identities* and *credentials* of the parties involved in the vehicular network operation. In general, authorities can be *multiple* and *distinct* in their roles and have a 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 online authorities is not required, as connectivity and communication, especially over a 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 also be meaningful in general.

In the context of secure VC, we interchangeably call authorities *Certification Authorities (CAs)*. Each CA is responsible for a *region* (national territory, district, county, etc.) and the identities and credentials of all the nodes registered from that region. To enable interaction between nodes from different regions, CAs provide certificates for other CAs (cross-certification) or provide *Foreigner Certificates (FCs)* to vehicles that are registered with another CA when they cross the geographical boundaries of their region (Section 5.6).

5.4.2 Node identification

At a basic level, we consider a network node, a vehicle or an infrastructure node, as: (a) a unique identity, V, (b) a public/private key pair K_V, k_V, (c) a module implementing the networking and the overlying application protocols, and (d) a module providing communication across a wireless network interface.

Each node is registered with only one CA, and has a unique *long-term* identity and a pair of *private* and *public* cryptographic keys. Accordingly, it is equipped with a long-term *certificate*. A list of *node attributes* and a *lifetime* are included in the certificate that the CA issues upon node registration and upon certificate expiration.

The binding of K_V to V and the binding of K_V to other data or *attributes* pertinent to V can be 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. Similarly, infrastructure nodes have a unique identity, I, and k_I and K_I private and public keys, with $Cert_Z\{K_I, A_I\}$ a certificate issued by an authority Z for I with attribute list A_I.[1]

Note that infrastructure nodes are not necessarily static. Vehicles can be grouped into two categories, *public* and *private*. The former can include vehicles related to public safety (e.g. highway assistance or firefighting, and police vehicles or helicopters), or public transportation vehicles (e.g. buses or trams). Public vehicles, like infrastructure nodes, are considered more trustworthy, and they can be used to assist security-related operations.

The CAs are also responsible for the *eviction* of nodes or the *withdrawal* of compromised cryptographic keys. This is achieved by revoking the corresponding certificates. In all cases, the interaction of nodes with the CAs is infrequent and intermittent, with the road-side infrastructure acting as a gateway to and from the vehicular part of the network, and the use of other infrastructure (e.g. cellular) also being possible.

5.4.3 Trusted components

Nodes are equipped with Trusted Components (TCs) (i.e. built-in hardware and firmware) that basically have two types of functionality: *cryptographic* operations and *storage*, in order to protect the vehicle's cryptographic material data (usable for liability identification). The TCs enforce a policy on interaction with the on-board software, including access to 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. The TCs should be *tamper-resistant*, in order to provide enhanced protection of the cryptographic material and other data.

A specific example of a TC is that of the Hardware Security Module (HSM) that the SeVeCom architecture envisions for vehicles and RSUs. The HSM stores and physically protects sensitive information (primarily private keys for signature generation) and it provides

[1] Users of VC systems can accordingly have a unique identity, U, and they can be bound to their credentials and secrets. They can be owners and/or the driver or passengers of vehicles, associated in general in a *many-to-many* manner. This chapter does not elaborate further on the user–vehicle interactions.

a secure time base. If a HSM were to be tampered with, for example to extract private keys, the physical protection of the unit would ensure that the sensitive information would be erased so that the adversary could not obtain it. Moreover, since all private key cryptographic operations are performed in the HSM, sensitive information never leaves the physically secured HSM environment. Essentially, the HSM is the basis of trust; without it, private keys can be compromised and their holders can masquerade as legitimate system nodes.

5.4.4 Secure communication

The basic way for nodes to undertake *secure communication* is for them to sign messages digitally, after attaching a time-stamp and the signer's location and certificate to the message. This way, modification, forgery, replay, and relay attacks can be defeated. The latter relate to secure neighbor discovery, which is possible since safety beacons include the time and location at the point they are sent across the wireless medium, as explained in Section 5.5.2. Signatures can be applied in different ways, to beacons or to multi-hop flooded and position-based multi- or uni-casted messages, not only by the message originator but also by relaying nodes (Section 5.5.3).

A conceptual view of a node is given in Figure 5.1. To provide both security and a degree of anonymity, long-term keys and credentials are *not* used to secure communication. Rather, the approach of *pseudonymity* or *pseudonymous authentication* is used. Each vehicle is equipped with multiple certified public keys (pseudonyms) that do not reveal the node identity. It obtains these pseudonyms via a trusted third party, a Pseudonym Provider (PNP), by proving it is registered with a CA. Then the vehicle uses each pseudonym and private key for at most τ seconds (the pseudonym lifetime), before it switches to another, not previously used, pseudonym. Messages signed under the same pseudonym can be trivially linked, but messages signed under different pseudonyms cannot (Sections 5.5.1 and 5.5.4).

5.5 Secure and Privacy-enhancing Vehicular Communication

5.5.1 Basic security

Periodic single-hop broadcasting, *beaconing*, is typically used for the so-called cooperative awareness applications: beacons, typically sent γ times per second, contain information on the sender's status such as vehicle position, speed, and heading. The frequency of beacons is expected to range from 10 Hz to 1 Hz. Beacon messages are digitally signed, and the signer's certificate is attached. More precisely, after the beacon message assembly is complete and before submitting a message m to the data link layer for transmission, the sending node (V) calculates a signature.

Rather than using its long-term cryptographic material, each node V is equipped with a set of *pseudonyms* (i.e. *public keys* that do not carry any information that identifies V). For the ith pseudonym K_V^i for node V, the CA provides a certificate $Cert_{CA}(K_V^i)$ that is simply a CA signature on the public key K_V^i (unlike, for example, the more complex X.509 certificate). The node uses the private key k_V^i corresponding to the pseudonym K_V^i to digitally sign messages. We term this approach the Baseline Pseudonymous Authentication (BPA) scheme, since it is essentially the same across different projects. Note that we should consider

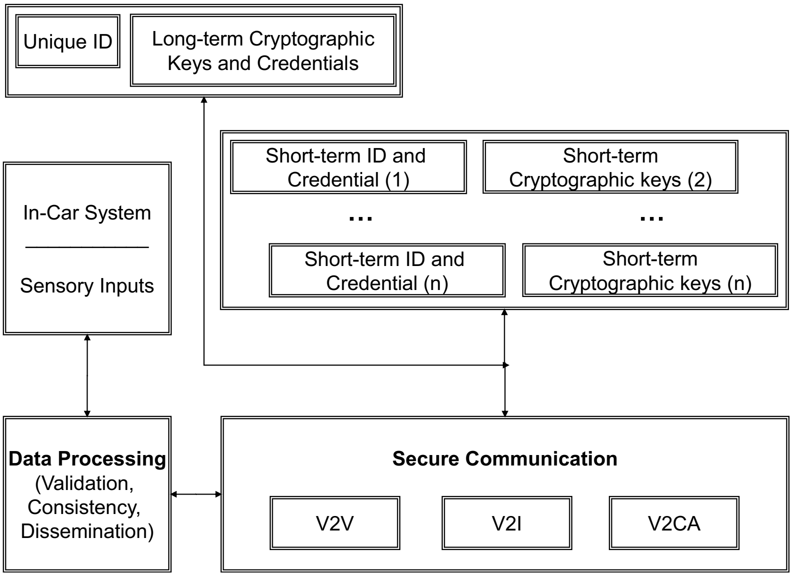
Unique ID
Long-term Cryptographic Keys and Credentials
In-Car System
Sensory Inputs
Short-term ID and Credential (1)
Short-term Cryptographic keys (2)
...
...
Short-term ID and Credential (n)
Short-term Cryptographic keys (n)
Data Processing
(Validation, Consistency, Dissemination)
Secure Communication
V2V
V2I
V2CA

Figure 5.1 Conceptual secure VC view: node functionality

only vehicles when using this approach, as the privacy of RSUs or other infrastructure does not need to be protected.

With $\sigma_{k_V^i}$ denoting V's signature under its ith pseudonym and m the message payload, the message format is m, $\sigma_{k_V^i}(m)$, K_V^i, $Cert_{CA}(K_V^i)$. Upon receipt of this, a node, with the public key of the CA assumed available, validates $Cert_{CA}(K_V^i)$. It makes use of a *Certificate Revocation List (CRL)*, also assumed to be distributed to vehicles, as discussed in Section 5.6. If successful (i.e. the K_V^i is not included in the CRL and the CA signature on K_V^i is valid), the node validates $\sigma_{k_V^i}(m)$. The CA maintains a map from the long-term identity of V to the node's set of pseudonyms, $\{K_V^i\}$. If presented with a signed message, the CA can perform the inverse mapping and identify the signer.

Each pseudonym is used for at most a period τ and is then discarded. A number of possible implementation aspects should be considered: the dynamic adaptation of the period of pseudonym usage; the number of pseudonyms (and the corresponding certificates and private keys) with which to 'pre-load' V; the frequency of such refills; other policies for pseudonym change, such as factors rendering a pseudonym change unnecessary (e.g. a TCP connection to an access point); and interactions of pseudonym changes with the network stack (Papadimitratos et al. 2007).

5.5.2 Secure neighbor discovery

Vehicles transmit safety beacons at high rates, and in this way they obtain a frequently updated view of other vehicles nearby (i.e. *physical neighbors*). This is the essence of cooperative awareness, which allows vehicles to have an up-to-date view of their neighborhood for transportation-related actions (e.g. avoidance of collisions).

However, it is often important that vehicles also discover other nodes (vehicles or RSUs) that are directly reachable (i.e. their *communication neighbors*) (Papadimitratos et al. 2008d). Typically, it is assumed that if two nodes are communication neighbors then they are physical neighbors, and vice versa, but this is not always the case, because adversaries mount *relay attacks*, by receiving and quickly retransmitting (replaying) messages from remote nodes.

The inclusion of sender time-stamp and location, along with authentication, enables the system to perform provably secure neighbor discovery against *external* adversaries (Poturalksi et al. 2008a,b). The basic idea is to estimate the sender–receiver distance based on a node's coordinates, the location in the received message, and the time-of-flight (difference between the node's time and the message time-stamp). Of course, to obtain precise results it is necessary to have time-stamps added by the hardware, and to have a cryptographic protection for the time-stamp calculated exactly at that point when the beacons are outgoing from the sender's transmitter. For a protocol-selectable acceptable neighbor range, the receiving node accepts the sender as a communication neighbor when the two distance estimates are equal and the sender is authenticated. As a result, vehicles can be assured that their *neighbor table* includes only nodes that are indeed communication neighbors.

5.5.3 Secure position-based routing

In order to disseminate data to a geographically defined destination, position-based communication benefits from location-aware nodes. Nodes maintain the locations of their neighbors, and forward data to the closest neighbor to the destination region. This approach is well suited for VC systems, given that location information is assumed to be available (e.g. by GNSS, for example the GPS). However, it can be abused by adversaries. As a basic security measure for both position-based routing and message distribution, source nodes sign created messages and attach the corresponding certificate, which is similar to the basic security functionality. Moreover, forwarding nodes can also sign packets they relay, so that these can be authenticated by the next-hop relay (Harsch et al. 2007). This way, only qualified network participants can create messages that are accepted by others, and message integrity is protected towards the destination. Replay and neighbor discovery attacks can be prevented, as discussed in the previous subsections. However, the location information in safety beacons can be forged by adversaries that seek to attract traffic illegitimately. A position verification scheme based on plausibility heuristics is capable of detecting such position falsifications (Leinmüller et al. 2006). More generally, tests that take into account constraints of the transportation network and the vehicle dynamics, and of course secure neighbor discovery, can be used for nodes to maintain a truthful neighborhood view (Festag et al. 2009).

5.5.4 Additional privacy-enhancing mechanisms

As the BPA scheme requires that the short-lived certificates are preloaded to the vehicle, a series of problems arise. What should a vehicle do when all of its pseudonyms are used up? How should the refilling of pseudonyms be designed and secured? What are the storage requirements, in terms of disk space and security? As the lifetime of a pseudonym is inversely analogous to the degree of message unlinkability, the stronger the protection needed the higher is the number of temporary identities and keys (pseudonyms) per node. For large-scale systems, this can be a significant burden.

To enhance system usability and efficiency, a method that allows nodes to self-generate (i.e. self-certify their own pseudonyms) was proposed by Calandriello et al. (2007) and Papadimitratos et al. (2008b). This approach extends the BPA, offering significant advantages: vehicles do not need to be sidelined or to compromise their users' privacy if a 'fresh' pseudonym is no longer available, no 'over-provisioning' in the supply of pseudonyms is necessary, and the cost of obtaining new pseudonyms over an 'out-of-band' channel is avoided.

This can be achieved with the use of *anonymous authentication* primitives, notably *Group Signature (GS)* as described in Section 5.5.4. As the practicality of GSs in the VC context is limited by their overhead, in terms of computation and communication, we propose in Section 5.5.4 the *Hybrid Pseudonymous Authentication (HPA)* scheme that allows on-the-fly generation of pseudonyms by combining the BPA and GS approaches. This alleviates the management overhead of the BPA; but in principle it is more costly than BPA. The mechanisms discussed in Section 5.5.5 reduce the cost of the HPA scheme to be roughly the same as that of BPA and increase the robustness of any pseudonymous approach.

Anonymous authentication: Group Signatures

Each node V is equipped with a secret *group signing key* gsk_V; the *group* comprises as members all vehicles registered with the CA. A *group public key* GPK_{CA} allows for the validation (by any node) of any *group signature* $\Sigma_{CA,V}$ generated by a group member. Intuitively, a group signature scheme allows any node V to sign a message on behalf of the group *without* V's identity being revealed to the signature verifier. Moreover, it is impossible to link any two signatures of a legitimate group member. Note that no public key or other credentials need to be attached to an anonymously authenticated message; the format is: $m, \Sigma_{CA,V}(m)$. Group signatures were introduced by Chaum and van Heyst (1991), with numerous subsequent works (e.g. Ateniese and Tsudik 1999; Boneh et al. 2004; Brickell et al. 2004; Syverson and Stubblebine 1999). If the identification of a signer is necessary, the CA can perform an *Open* operation and reveal the signer's identity (Bellare et al. 2003, 2005).

Hybrid Pseudonymous Authentication

The combination of the BPA and GS schemes is the basic element of the proposal of Calandriello et al. (2007) and Papadimitratos et al. (2008b). Each node V is equipped with a group signing key gsk_V and the group public key GPK_{CA}. Rather than generating group signatures to protect messages, a node generates its own set of pseudonyms $\{K_V^i\}$. As in Section 5.5.1, a pseudonym is a public key without identification information, and $\{k_V^i\}$ is the set of corresponding private keys. In this case, the CA does not provide a certificate on K_V^i, but V uses gsk_V to generate a group signature $\Sigma_{CA,V}()$ on each pseudonym K_V^i instead.

This way, nodes generate and 'self-certify' K_V^i on the fly, producing $Cert_{CA}^H(K_V^i)$. The H superscript denotes the HPA scheme and differentiates this certificate from that of the BPA approach. The *CA* subscript denotes that the certificate was generated by a legitimate node registered with the CA. V attaches the $Cert_{CA}^H(K_V^i)$ to each message, and signs with the corresponding k_V^i: $m, \sigma_{k_V^i}(m), K_V^i, Cert_{CA}^H(K_V^i)$.

When receiving an HPA message, the group signature $\Sigma_{CA,V}(K_V^i)$ is verified, using the GPK_{CA}. If this is successful, the receiver infers that a legitimate system (group) member generated pseudonym K_V^i. We emphasize that, as per the properties of group signatures, the receiver/verifier of the certificate *cannot* identify V and *cannot* link this certificate and pseudonym to any prior pseudonym used by V. Once the legitimacy of the pseudonym is established, the validation of $\sigma_{k_V^i}(m)$ is identical to that of the BPA message. To identify the signer of message, an *Open* on the $Cert_{CA}^H(K_V^i)$ group signature is necessary; the message m is bound to K_V^i via $\sigma_{k_V^i}(m)$, and K_V^i is bound to V via $\Sigma_{CA,V}(K_V^i)$.

5.5.5 Reducing the cost of security and privacy enhancing mechanisms

Mechanisms have been proposed in the literature to reduce overhead (Mechanisms 1, 2, and 4 below) and enhance robustness (Mechanism 3). They are all applicable to both BPA and HPA schemes. To reduce overhead, Calandriello et al. (2007) and Papadimitratos et al. (2008b) propose *not* to attach certificates to all messages, but rather to do so for one in every α successive beacons; they also propose certificate caching to reduce the verification processing overhead. Moreover, Kargl et al. (2008b) propose to avoid attaching certificates to beacons *unless* a change in the vehicle neighborhood is detected.

Mechanism 1

At the sender side, the $Cert_{CA}^H(K_V^i)$ is computed only once per K_V^i, because $Cert_{CA}^H(K_V^i)$ remains unchanged throughout the pseudonym lifetime τ. Note that the notation here does not distinguish which method is used for the certificate generation. For the same reason, at the verifier's side the $Cert_{CA}^H(K_V^i)$ is validated upon the first reception and stored, even though the sender appends it to multiple (all) messages. For all subsequent receptions, if the $Cert_{CA}^H(K_V^i)$ has already been seen, the verifier skips its validation. This optimization is useful because $\tau \gg \gamma^{-1}$ (i.e. the pseudonym lifetime is much higher than the beaconing period).

Mechanism 2

The sender appends its signature $\sigma_{k_V^i}(m)$ to all messages, but it appends the corresponding K_V^i, $Cert_{CA}^H(K_V^i)$ only once every α messages (termed the *certificate period*). The message structure is m, $\sigma_{k_V^i}(m)$. To make the choice of the right K_V^i to verify such an incoming message easy, all messages signed under the same pseudonym can carry a short key ID field. When a pseudonym change occurs, the new tuple $\sigma_{k_V^{i+1}}(m)$, K_V^{i+1}, $Cert_{CA}^H(K_V^{i+1})$ must be computed and transmitted. V will sign messages with the new k_V^{i+1} corresponding to K_V^{i+1} from then on.

Mechanism 2 can affect the protocol robustness, if the message that carries K_V^{i+1} and $Cert_{CA}^H(K_V^{i+1})$ is not received. Then, nodes in range of V must wait for α messages before the next pseudonym transmission, while being unable to validate *any* message from V. This can be dangerous if vehicles are close to each other and/or moving at high relative speeds.

Mechanism 3

To address the aforementioned issue with Mechanism 2, the transmission of K_V^{i+1}, $Cert_{CA}^H(K_V^{i+1})$ is repeated for β consecutive messages when K_V^{i+1} is issued, with β denoted as the *push period*.

Mechanism 4

The K_V^{i+1}, $Cert_{CA}^{H}(K_V^{i+1})$ is transmitted (repeated) only when V detects a new neighbor, so that transmissions from V can be validated by the 'newcomer'. Mechanism 3 can be combined with Mechanism 4 to enhance reliability.

Cryptographic overhead and system performance

Cost constraints in today's car manufacturing make it hard to equip vehicles with powerful state-of-the-art desktop processors. Instead, relatively inexpensive and energy-saving embedded processors are used. But cryptographic operations create a significant overhead, both in terms of processing and of communication bandwidth, especially because vehicles send information frequently (e.g. position and environment conditions) – typically one beacon per 100 milliseconds (i.e. $\gamma = 10$).

Without ignoring other factors, the computational security overhead is due to the generation and verification of packet signatures and certificates. The communication security overhead is due to signatures and certificates attached to packets. Each safety beacon has to be signed, and each vehicle has to validate (for example, every 100 milliseconds) beacons from *all* neighboring vehicles in range, which, do not forget, may also change their identity (pseudonym) in the meantime.

Cryptographic and communication security overhead can affect VC applications in multiple ways. The first dimension of the problem is communication reliability: increased beacon size contributes to interference. In principle, the higher the offered load, with the number of transmitters in the area, the beaconing rate, and the message overhead, the worse the channel performance. The second dimension is processing overhead. Attention should be paid to the cost of verifications: safety beaconing, for example, entails that each node verifies at least one signature for each received packet within the beaconing period, with all its neighbors sending one beacon, and it has to generate one signature during the same period.

The mechanisms discussed above can significantly reduce both communication and processing overhead. This can be beneficial in any case, as long as the supported applications are not affected. Any safety-related warning can be trusted only if it can be cryptographically validated. For example, a safety beacon can be validated only if the corresponding short-term certified public key (pseudonym) of the signer was previously verified. A specific safety application is considered by Calandriello et al. (2007) and Calandriello et al. (2009); Papadimitratos et al. (2008b): an emergency braking warning application operating on top of secure and privacy-enhancing communication, which can be almost as effective as the same application operating without any security (or related overhead).

5.6 Revocation

All projects on security architectures for VC consider the eviction of faulty or illegitimate nodes. More generally, certificates of faulty or compromised nodes should be revoked. Without valid credentials, faulty or adversarial nodes can no longer damage the VC system. Nodes are revoked in principle in three cases: if they are deemed faulty, if it is detected that their cryptographic keys have been compromised, or for administrative reasons. The CA is responsible for a revocation decision. If the decision is the result of a (detected) faulty operation or key compromise, then the CA should obtain or be presented with evidence. One option, which relieves the CA from operating its own monitoring infrastructure, is to have misbehavior evidence collected by vehicles.

The basic revocation approach, as is the case for other systems beyond VC, is through the distribution of Revocation Lists (RLs) that the CA generates and authenticates. At a first stage, in a system that relies on pseudonyms (short-term identities) and corresponding private keys, the CA (or PNP) would not provide new pseudonyms to a revoked node. Nonetheless, the use of a RL would be necessary for the revocation of the pseudonyms that are still valid. The RL can be a list of certificate identifiers, similar to certificate revocation lists (CRLs) for classic public key cryptography; or it can be a list of elements that allow for the identification of the signer when group signatures are used (e.g. as in a GS scheme, when HPA is used). For the rest of the discussion, we do not dwell on the exact type of RL; a discussion about some quantitative aspects is available in Calandriello et al. (2009).

The basic challenge is to distribute a RL efficiently and effectively across a large-scale multi-domain system, illustrated by Figure 5.2. This can be achieved, in spite of the constraints of the VC environment, by leveraging on a sparse road-side infrastructure. A scheme proposed by Papadimitratos et al. (2008c) achieves that with very low bandwidth used for RL transmissions, on the order of a few kbit/s at each RSU. In practice, with RSUs a few kilometers apart, all vehicles can obtain the latest RL within tens of minutes (e.g. the duration of a commuter journey). *Scalability* can be achieved by keeping RL sizes low and RSU–CA interactions minimal, with no RSU–RSU interactions.

This scheme relies on few basic elements. Thanks to the *collaboration between regional CAs*, RLs contain only regional revocation information and their size is kept low. *Encoding* of RLs into numerous (cryptographically) self-verifiable pieces provides resilience to disconnections, radio impairments, and malicious message injection.

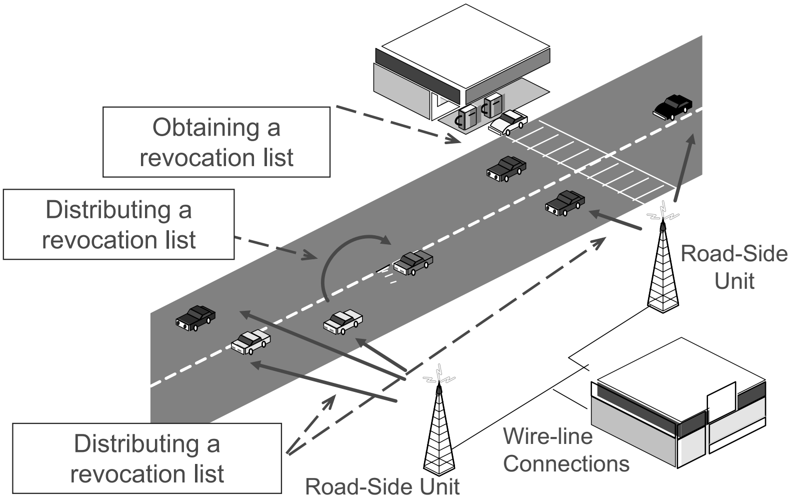
Obtaining a revocation list
Distributing a revocation list
Road-Side Unit
Distributing a revocation list
Wire-line Connections
Road-Side Unit

Figure 5.2 Illustration of revocation list distribution

The multi-domain CA structure keeps RL sizes low, but vehicles need revocation information from other regions to validate the certificates of foreigner (visiting) vehicles.

Rather than distributing RLs of other regions, the CA validates certificates of visiting nodes: if they are not revoked in their home region, it issues them short-lived foreign certificates (*FCs*) that they must use in the foreign region. If the certificate of a FC holder is revoked later on, it is included in the RL of the CA that issued the FC, and its actual certificate is added to its home CA's RL.

The encoding of the RL into multiple pieces can be done in different ways, using *Fountain* or *Erasure* codes. The original RL is segmented into M parts and then encoded, with added redundancy. Erasure codes produce $N > M$ *RL pieces*, such that for any M out of N pieces received, the original RL can be reconstructed (Rabin 1989). Fountain codes and among them a special class, Raptor codes, with linear time encoding and decoding complexity, produce for M input pieces a potentially limitless stream of output RL pieces (Shokrollahi 2006). For a protocol-selectable parameter $\sigma > 0$, the original M pieces can be recovered with a high probability from any subset of $M(1 + \sigma)$ RL pieces. The RL version and time-stamp, a piece sequence number, the CA identifier, and a digital signature covering all previous fields, are added to each RL piece, so that each of them can be validated individually.

For areas that are not covered by RSUs, vehicles could undertake the role of distributing the RL in an '*epidemic*' manner (Laberteaux et al. 2008; Papadimitratos et al. 2008a). It would also be possible to use alternative media for such information, such as cellular links (e.g. General Packet Radio Service, or GPRS, or Universal Mobile Telecommunications System, or UMTS), broadcast digital radio, or simply localized wireless or wired links when the vehicle is static (e.g. overnight parking).

If the adversary could not control the communication of the CA and the on-board trusted hardware, all credentials and cryptographic keys could be remotely removed by the CA. This can be achieved by a 'kill' command issued to the HSM: once it authenticates the command, the HSM erases all its private keys (Papadimitratos et al. 2008a; Raya et al. 2007). This essentially prevents the revoked node from participating further in the protocol execution: its messages cannot be signed and validated.

The decision to revoke or not the credentials of a node is made exclusively by the CA. Nonetheless, CRLs would be issued infrequently – for example, once per day or every few days. This would leave a vulnerability window, until a faulty or otherwise compromised node's credentials are revoked. A local reaction mechanism can protect honest nodes from a misbehaving node that is not yet revoked. This, of course, presumes that nodes can reliably detect a misbehavior and attribute it to a node. One option is to have each node that detects a wrongdoer broadcast a warning in its neighborhood. When multiple warnings are issued and corroborated locally for a given node, then newcomers in the vicinity of the detected attacker can ignore its messages (Raya et al. 2007).

Clearly, the redundancy in detection information can be beneficial: it can reduce false positives (i.e. 'branding' a correct or honest node as a misbehaving one). Similarly, an attacker that issues false warnings by trying to exclude honest nodes cannot achieve anything if it acts alone. Or, more generally, multiple attackers would need to be present together in a neighborhood, with correct nodes being the minority, in order to have honest nodes excluded. On the flip side, the distributed computation can be slow to complete, and can be affected significantly by the high mobility, or incur higher communication costs in a dense topology. An alternative method, which is complementary, would be to allow a single node to announce any misbehaving nodes it detects (Moore et al. 2008). This, of course, would be less robust against attackers that abuse the misbehavior reporting mechanism. But, overall, it would allow for a faster reaction in terms of detecting attackers.

5.7 Data Trustworthiness

Vehicular communication systems are *data centric*: (a) the VC-enabled on-board system depends on many sensory inputs; (b) frequent data and event reports are exchanged among vehicles and road-side infrastructure; and (c) the identity of the sender of such data is of lesser importance. The sensory inputs vary in nature, ranging from vehicle-specific data (e.g. motion or temperature sensors) to location and time corrections, provided, for example, by GNSS. Transportation safety and efficiency applications are built on the exchange of data. Safety beacons carry the location (but can also carry other information such as speed and direction) of the transmitting vehicle; dangerous conditions or re-routing warnings can be broadcast by infrastructure and vehicles. For all such messages, the identity of their sender is not important in the way that an IP address is for other networks. On the contrary, information such as the time and location of the reporting node, as well as its attributes (e.g. type of vehicle or RSU), are the important information, along with the reported data *per se*. Finally, privacy-enhancing mechanisms conceal the identity of the vehicles.

Cryptographic protection, including misbehavior detection and node eviction, addresses a significant part of the problem: it prevents external adversaries from injecting bogus data. External adversaries could still affect sensory inputs, with attacks remaining undetected in the absence of additional mechanisms, whereas internal adversaries can inject any bogus data at will. Only when they are detected can they be isolated and eventually evicted (Section 5.6). But interacting with possibly adversarial (faulty) data senders in order to determine their trustworthiness is not easy. Adversaries can intelligently change their attack patterns (e.g. remaining below the 'detection radar' most of the time but still harming the system). Moreover, detection implies a lengthy interaction, which is often impossible to sustain; encounters are in general short lived and without prior association.

Security mechanisms that allow nodes to detect and ignore bogus data are necessary. In general, nodes cannot rely only on their own measurements or have access to trusted data, and data often come from remote sources. Each node should be able to assess their trustworthiness alone. The use of *non-cryptographic* protection mechanisms is paramount. In the rest of this section we consider two cases: location information and data from other VC-enabled applications.

5.7.1 Securing location information

Location information is critical for VC systems, especially for cooperative awareness, vehicle collision avoidance and essentially all safety applications, as well as for position-based information dissemination. GNSS, such as the GPS, its Russian counterpart (GLONAS), and the upcoming European GALILEO system, are the most widely used technologies: GNSS transmit signals bearing reference information from a constellation of satellites; computing platforms, equipped with the appropriate receiver, can decode them and determine their own location. Most importantly, these units are already integrated in vehicles or available in large numbers, as part of commodity devices for navigation.

However, commercial instantiations of GNSS are open to abuse: according to a recent article (Humphreys et al. 2009), software-defined GPS receivers make spoofing (i.e. the injection of forged navigation messages by an adversary, see Figure 5.3) relatively easy: the hardware can be assembled with off-the-shelf components. In the same article, an effective spoofer is presented. With such an ability, the adversary can influence the location information, $loc(V)$, that a node V calculates, and compromise the node's operation. For example, in the case of a fleet management system, an adversary can target a specific truck. First, the adversary can use a transmitter of forged GNSS signals that overwrite the legitimate GNSS signals and are received by the victim node (in this example, the truck) V. This would cause a false $loc(V)$ to be calculated and then reported to the fleet center, essentially concealing the actual location of V from the fleet management system. Once this is achieved, physical compromise of the truck (e.g. breaking into the cargo or hijacking the vehicle) is possible with a reduced or no ability for the system to detect it and react in time.

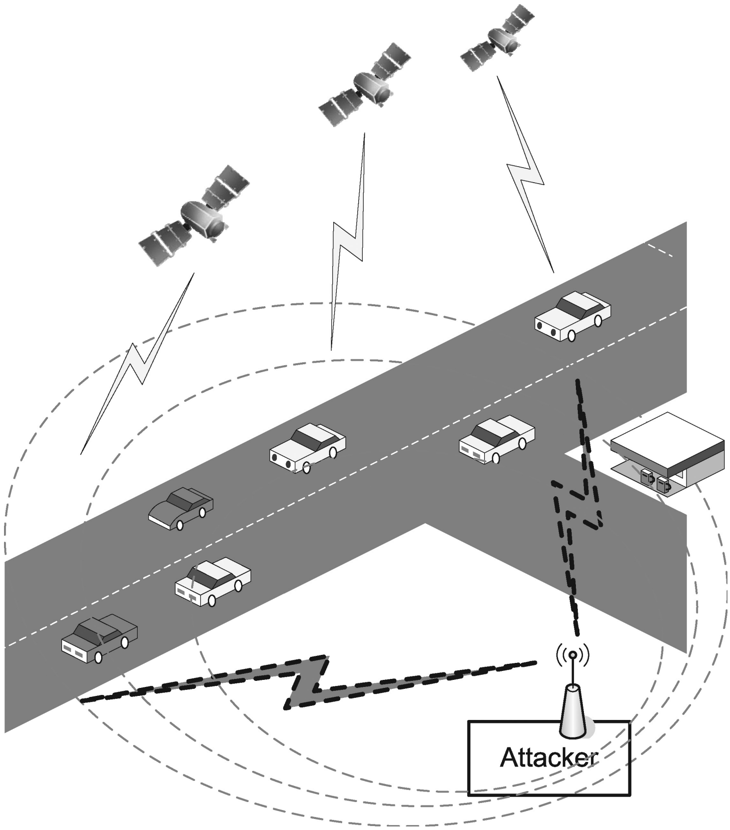

Figure 5.3 Illustration of an attack against GNSS-based localization, in which the attacker forges or replays GNSS signals

It is useful to repeat here a long-known fact: location information in the VC system cannot be considered trustworthy by default. Spoofers fall into the category of input-controlling adversaries (Section 5.2), and such attacks are possible without compromising, physically or otherwise, the GNSS receiver or other on-board equipment or software. A variant of such attacks, termed replay attacks, would be possible even if the GNSS were cryptographically protected, as is expected for the authentication services of the upcoming GALILEO system. Replay attacks can be fine grained, so that the gradual manipulation of the victim's location can remain small and thus hard to detect. But, cumulatively, they can lead to substantial distances between the actual and the perceived (provided by the GNSS) location of the victim nodes (Papadimitratos and Jovanovic 2008b).

Defense mechanisms, complementary or even orthogonal to cryptographic protection, are a promising, low-cost, and effective approach. Three defensive mechanisms that allow receivers to detect forged GNSS messages and to fake GNSS signals have been developed and analyzed by Papadimitratos and Jovanovic (2008a). The countermeasures rely on information that the receiver obtained before the onset of an attack, or, more precisely, before the suspected onset of an attack. This can be (a) a node's own location information, calculated from GNSS navigation messages; (b) clock readings, without any re-synchronization with the help of the GNSS or any other system; and (c) received GNSS signal Doppler shift measurements. Based on these different types of information, it can be detected whether the received GNSS signals and messages are genuine or whether they originate from adversarial devices. If the latter, location information induced by the attack can be rejected, and manipulation of the location-aware functionality be avoided.

5.7.2 Message trustworthiness

The classic approach is to assess the trustworthiness of messages sent by a node (vehicle or RSU) primarily based on the trustworthiness of the sender's credentials. CAs (and nodes when they issue their own pseudonyms) make statements about public keys, identities, and attributes, and special types of data (e.g. revocation lists). Nodes (vehicles and RSUs) make statements about various types of data. At any point in time, messages from any newly encountered nodes are trusted as long as their certificates are valid. These entity-centric trust relations, which are set *a priori*, are useful, but they do not provide the necessary flexibility for the highly volatile and data-centric VC systems.

In VC systems, unlike traditional trust establishment schemes, the data trust level cannot be the same as that of the data-generating node. Clearly, different types of nodes can have by default different levels of trustworthiness; for example, police cars are more trustworthy than private cars. But the trust level of reports from the same type of vehicle can also vary: reports issued at different distances from the corresponding events, in terms of location or time, or the timeliness of the report itself, should have their trust values adapted accordingly. The receiver of such reports could assign the same trust level to those coming from different vehicles but referring to the same event. Moreover, it could use such multiple pieces of evidence to corroborate the event, and then issue its own report. In these cases, it is also clear that the report (data) trust values would differ from those of the reporting vehicle(s).

In such a context, it is far more useful to assess the trustworthiness of data *per se*, rather than to rely on the fixed node trust levels. Raya et al. (2008) present a scheme to instantiate this approach. Default trust levels are used, but only as one factor. In addition, dynamic factors (location, time, number, and type of the statements on data) are used to calculate on the fly the trustworthiness of data, or more essentially the truthfulness of the reported event. This assessment is based on multiple sources of evidence, each of which is assigned a weight calculated via well-established rules. All pieces of weighted evidence and their weights are combined in a decision logic (such as Dempster–Shafer Theory, Bayesian inference, or majority voting) that accounts for the uncertainty of the data.

5.8 Towards Deployment of Security and Privacy Enhancing Technologies for Vehicular Communication

5.8.1 Revisiting basic design choices

Is security always necessary and meaningful?

Is an arguably complex secure VC system actually necessary? This is a legitimate question, given that cellular telephony and nomadic wireless Internet access were deployed without strong security features. They proliferated and continue to do so, in spite of significant security and privacy breaches. But, the significant differences of VC systems from these two systems imply that a different approach is necessary. First, cellular and nomadic network access rely on infrastructure, which simplifies the provision of security; for example, associations (trust) are needed between the mobile node and the infrastructure. Moreover, a compromise would not have costly or even fatal consequences, such as a multi-car accident caused by an attack on a VC system.

The stakes for VC systems are higher: reducing accidents, saving lives, improving transportation. Thus, a single security incident would perhaps suffice for the public to lose confidence in this new technology. Then, assuming sufficient security is in place, one can ponder on the *degree of protection* a security architecture should and can offer. Would strong cryptographic protection of network and application protocols suffice to ensure that false data are not injected into the system? Given the rich location information on VC traffic (e.g. in safety beacons), how easy is it for an adversary to recreate trajectories of vehicles even if their transmissions are fully anonymous (e.g. utilizing GS or BPA, with each message signed under a different temporary identity)? This is feasible in an area that is fully covered by the adversary, but the adversary could also derive and use additional knowledge, so as to be effective even when it cannot intercept VC in some areas (Buttyan et al. 2007). Would anonymous or pseudonymous secure VC ensure location privacy when numerous cameras and optical plate recognition systems are deployed?

VC security does not address problems that are present independently of the use of VC. But it fends off a broad range of exploits that could otherwise wreak havoc with VC and the transportation system. External adversaries or vehicle modifications are mitigated, the results of key compromise are thwarted, and accountability, even if anonymous authentication is used, can lead to the eviction of adversarial nodes. Until this happens, redundancy or the absence of corroborating evidence from other nearby vehicles could enable the degree of truthfulness of received VC application messages to be inferred. Initial results are promising, showing that *data-centric trust establishment* is feasible (Section 5.7.2). Investigations for specific applications and complex environments, as well as measures to thwart determined adversaries, can lead to stronger protection.

Choice of cryptographic tools

The high volatility and large scale of VC systems led to the choice of digital signatures. For the selection of appropriate algorithms, the following basic factors were considered: the processing times for signature generation and verification, the security overhead (public key and signature sizes), the standardization of cryptographic algorithms (and confidence in their strength), and the experience in implementation. Elliptic curve-based algorithms (e.g. Elliptic

Curve Digital Signature Algorithm, or EC-DSA) seem to be preferred, primarily because of low network overhead for strong security. Thanks to usability limitations, approaches such as the HPA (Section 5.5.4) enable on-board, on-the-fly generation of short-term keys and credentials. However, the use of anonymous authentication for this purpose should prevent any abuse: adversaries should not be allowed to use anonymity to influence or control VC protocols and applications. Detection of multiple uses of anonymous transmissions, beyond a threshold within a given period of time, perhaps by revealing the identity of the wrongdoer, is a necessary addition (Camenisch et al. 2006).

Security levels for VC system entities have not been clearly defined yet, but 80-bit or higher security seems to be favored, to prevent practical cryptanalytic attacks. It is important, though, to consider *for which operation* a specific security level for cryptographic primitives is needed. Clearly, CAs and PNPs should have higher security levels. Then high security would be needed for long-term keys and certificates. The lowest level should be assigned to short-term keys, for which security levels below 80 bits could perhaps be considered. Even if such a key, valid for minutes or hours, were broken within weeks or months by a determined adversary, there would be no immediate consequences, only a reduction in overhead. Of course, this would be true if and only if VC traffic were not used in the long run – for example, logged for future liability attribution (Section 5.8.2). In all cases, as discussed in Section 5.5.5, sufficient processing power for the employed cryptographic functionality should be provided to ensure the overall application performance that is sought.

Trustworthy Vehicular Communication equipment?

VC system nodes can often have low physical protection. If vehicular equipment is indeed trustworthy, with the appropriate VC security in place, the overall problem of securing VC systems could be more easily addressed. There are critical resources to be protected: A TC, for example, such as the HSM proposed by SeVeCom, would store private keys and perform private cryptographic operations. With a tamper-resistant TC, extraction of the private keys would be impossible. With a real-time clock and a battery integrated in the TC, the adversary would be unable to feed the TC with fake future time-stamps and obtain falsified cryptographically protected messages. However, cost is a major concern, and making all of the on-board equipment tamper-proof or tamper-resistant would be impractical.

What type of revocation?

The distribution of RLs, discussed in Section 5.6, is the basic approach in VC systems. A RL can be a CRL, if traditional cryptography is used (e.g. for BPA), but it can differ in the case of the HPA scheme. Checking whether a node is in the RL can incur significant processing overhead, primarily because of anonymity mechanisms. For ‘classic’ cryptography, each pseudonymous certificate should be validated at first reception, but with many pseudonyms per vehicle, the RL would be large. For anonymous credentials, each received message should be checked against the much shorter RL, but each check is orders of magnitude costlier than that for ‘classic’ cryptography.

The challenge is not the RL distribution but rather its on-board processing cost, which is proportional to the RL length. The natural question to ask is whether it is necessary to ignore messages signed by all nodes in a RL. In fact, this is closely related to the composition of the RL. For example, if a stolen vehicle is in the RL, its VC equipment is not necessarily compromised; thus it would be unwise, in the interests of the safety of receiving vehicles, to ignore its messages. A flexible approach to address the problem could reduce the length of RLs and, thus, the processing overhead: distinct RLs are created, according to the 'urgency' of using them for real-time message validation. The RL of highest priority, processed at all times, can then contain only the truly faulty or compromised nodes. At a lower priority can be a RL with nodes that are possibly faulty, perhaps in different RLs according to the type of fault, and at the lowest priority would be a RL with nodes evicted for other reasons. Lower priority RLs can checked if possible, or if a specific event triggers the need to do so (e.g. suspected faulty behavior by a nearby node).

5.8.2 Future challenges

Introducing (secure) Vehicular Communication to the market

The development of VC systems can significantly influence security solutions and thus the system trustworthiness. The primary question is how the VC deployment would take place. Would it be based on '*all-in-one*' on-board equipment, or would it be based on a sequence of *add-on* components? In other words, would the OBU be one or two powerful, multi-purpose box(es), or perhaps a multi-core processor, running all protocols? Or would it be a set of boxes, each of them added on board gradually, running a single application with just enough processing power for the specific tasks?

An all-in-one model resembles what has been considered thus far in the development of secure VC architectures. But the add-on approach may be closer to what a strongly market-centric deployment commands, driven by the applications (e.g. entertainment) preferred by consumers. Reflecting the mindset of some stakeholders, the evolutionary deployment would most likely lead to a minimum application-specific security as well as a heterogeneous on-board network. The situation would become more involved if user devices (e.g. Personal Digital Assistants, or PDAs, cell-phones, and home or corporate computers) interact with the OBUs, for example to obtain useful personal information for navigation, to record trip data, or to access physical spaces or digital content. All of these aspects would raise new challenges in terms of security and privacy (Papadimitratos 2008).

Organizational concerns

The reliance on authorities is in line with long-lived approaches in managing vehicles (Papadimitratos et al. 2006b). However, the effort of operating CAs often results in a degree of skepticism, with frequently recurring questions on the operational cost or the difficulties of collaboration among diverse CAs. The alternative, of vehicle manufacturers running their own CAs, is being considered. Nonetheless, this raises concerns about monopoly or oligopoly situations that could be imposed, or even the likelihood that proprietary solutions that do not provide fully fledged security might be adopted. Existing *multi-domain* systems, such as cellular systems, which require access control and accounting, indicate that addressing organizational issues is feasible. In fact, the success story of cellular systems provides useful clues. Numerous distinct providers, each having high numbers of registered clients and devices, all uniquely identified and able to operate in other regions while being billed for network usage via their 'home' providers, is a model that has interesting features and even similarities to ponder.

Legal considerations

User awareness of the offered protection is paramount: the guarantees that the VC system offers, the residual vulnerabilities, and the role of all system entities should be clearly stated in end-user agreements. Analogies can be drawn with existing systems: for example, with recent privacy breaches against cellular telephony perpetrated by insiders. The responsibility of each entity, the users included, should be clear, as this also relates to VC equipment maintenance and accreditation.

The use of VC systems to assist in the attribution of *liability* for transportation incidents is a controversial issue. Clearly, a non-secure VC system would be out of the question for such a task. Strong accountability in secure VC systems, as discussed above, is possible. However, determining which entity can perform this, and under which circumstances, and then through which procedure liability could be attributed, is far from straightforward.

Policies for VC systems would also have to deal with the issue of *voluntary or mandatory* use of the equipment. Would, for example, safety and traffic efficiency functionality be mandatory, in the same way that seatbelts are in many countries nowadays? It is likely that users would have an incentive to run these applications, for example in order to lower their insurance premiums. But if deployment is mandatory, would privacy concerns be fully addressed? Solutions discussed above can indeed achieve this. But users may still raise legitimate arguments in favor of powering off their VC boxes. Or perhaps users may raise the need for distinct secure VC instantiations, for example for government vehicles that do not wish to take any risk of being traced by terrorists.

5.9 Conclusions

Significant progress has been made already towards comprehensive security- and privacy-enhancing solutions for VC systems. Moreover, the design of VC protocols is evolving and standardization efforts are ongoing. The research community has a unique opportunity: to understand the problems at hand deeply, and, at the same time, to design VC protocols and applications by taking into account security and PET mechanisms. For example, OBU characteristics can be set to a certain standard to enable security; or protocol features that enhance performance but allow high-impact attacks can be disabled for enhanced resilience.

At the same time, investigations of VC systems and their security reveal new dimensions and complexity. VC systems would interact with various other systems, essentially forming a wireless system of systems. Moreover, results and insights can be far reaching and applicable to other computing systems. The current extensive interest, rising awareness, and significant results in terms of security for VC systems, along with demonstrations, are most welcome (Ardelean and Papadimitratos 2008; Gerlach et al. 2008; Kargl et al. 2009). Nonetheless, further progress in securing all aspects of the VC system, and extensively evaluating the overall system performance through test beds, is necessary. The objective is to have trustworthy VC systems at the time of their initial deployment, so that societies can reap the benefits of intelligent transportation systems.

References

Ardelean, P. and Papadimitratos, P. (2008) Secure and Privacy-Enhancing Vehicular Communication Demonstration. *IEEE WiVec*, Calgary, AL, Canada.

Ateniese, G. and Tsudik, G. (1999) Group Signatures à la carte. *SODA '99*, Baltimore, MD, USA.

Bellare, M., Micciancio, D. and Warinschi, B. (2003) Foundations of Group Signatures: Formal Definition, Simplified Requirements and a Construction based on Trapdoor Permutations. *Advances in Cryptology*.

Bellare, M., Shi, H. and Zhang, C. (2005) Foundations of Group Signatures: The Case of Dynamic Groups. *CT-RSA*, San Francisco, CA, USA.

Boneh, D., Boyen, X. and Shacham, H. (2004) Short Group Signatures. *Crypto '04*, Santa Barbara, CA, USA.

Brickell, E., Camenisch, J. and Chen, L. (2004) Direct Anonymous Attestation. *CCS '04*, Washington DC, USA.

Buttyan, L., Holczer, T. and Vajda, I. (2007) On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs. *ESAS*.

Calandriello, G., Papadimitratos, P., Hubaux, J.P. and Lioy, A. (2007) Efficient and Robust Pseudonymous Authentication in VANET. *ACM VANET*, Montreal, Quebec, Canada.

Calandriello, G., Papadimitratos, P., Hubaux, J.P. and Lioy, A. (2009) On the Performance of Secure Vehicular Communication Systems. *LCA-REPORT-2009-006*.

Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A. and Meyerovich, M. (2006) How to Win the Clone Wars: Efficient Periodic n-Times Anonymous Authentication. *ACM CCS*, Alexandria, VA, USA.

Chaum, D. and van Heyst, E. (1991) Group Signatures. *EUROCRYPT '91*, Brighton, UK.

Festag, A., Papadimitratos, P. and Tielert, T. (2009) Design and Performance of Secure Geocast for Vehicular Communication. *LCA-REPORT-2009-007*.

Gerlach, M., Friederici, F., Ardelean, P. and Papadimitratos, P. (2008) Security Demonstration – C2C-CC Forum and Demonstration, Dudenhofen, Germany.

Harsch, C., Festag, A. and Papadimitratos, P. (2007) Secure Position-Based Routing for VANETs. *IEEE Vehicular Technology Conference (VTC2007-Fall)*, Baltimore, MD, USA.

Humphreys, T., Psiaki, M., Kintner, P., Ledvina, B. and O'Hanlon, B. (2009) Assessing the Spoofing Threat. *GPS World*.

IEEE 1609.2 (2006) Trial-Use Standard for Wireless Access in Vehicular Environments – Security Services for Applications and Management Messages.

Kargl, F., Ma, Z. and Schoch, E. (2006) Security engineering for VANETs. *Proceedings of the Fourth Workshop on Embedded Security in Cars (ESCAR)*, pp. 15–22, Berlin, Germany.

Kargl, F., Papadimitratos, P., Buttyan, L., Müter, M., Wiedersheim, B., Schoch, E., Thong, T.V., Calandriello, G., Held, A., Kung, A. and Hubaux, J.P. (2008a) Secure Vehicular Communications: Implementation, Performance, and Research Challenges. *IEEE Communications Magazine* **46**(11), 110–118.

Kargl, F., Papadimitratos, P., Holczer, T., Cosenza, S., Held, A., Mueter, M., Asaj, N., Ardelean, P., de Cock, D., Sall, M. and Wiedersheim, B. (2009) Secure Vehicle Communication (SeVeCom) Demonstration. *IEEE MobiSys*, Krakow, Poland.

Kargl, F., Schoch, E., Wiedersheim, B. and Leinmüller, T. (2008b) Secure and Efficient Beaconing for Vehicular Networks. *Proceedings of the Fifth ACM International Workshop on Vehicular Ad hoc Networks (VANET)*, San Francisco, CA, USA.

Laberteaux, K., Haas, J. and Hu, Y. (2008) Security Certificate Revocation List Distribution for VANET (short paper). *ACM VANET*, San Francisco, CA.

Leinmüller, T., Schoch, E. and Kargl, F. (2006) Position Verification Approaches for Vehicular Ad Hoc Networks. *IEEE Wireless Communication Magazine* **13**(5), 16–21.

Moore, T., Raya, M., Clulow, J., Papadimitratos, P., Anderson, R. and Hubaux, J.P. (2008) Fast Exclusion of Errant Devices from Vehicular Networks. *IEEE SECON*, San Francisco, CA, USA.

NoW (2007) Network On Wheels. URL: http://www.network-on-wheels.de/.

Papadimitratos, P. (2008) 'On the Road' – Reflections on the Security of Vehicular Communication Systems. *Proc. IEEE International Conference on Vehicular Electronics and Safety ICVES 2008*, pp. 359–363.

Papadimitratos, P. and Jovanovic, A. (2008a) GNSS-based Positioning: Attacks and countermeasures. *Proc. IEEE Military Communications Conference MILCOM 2008*, pp. 1–7.

Papadimitratos, P. and Jovanovic, A. (2008b) Protection and fundamental vulnerability of GNSS. *Proc. IEEE International Workshop on Satellite and Space Communications IWSSC 2008*, pp. 167–171.

Papadimitratos, P., Buttyan, L., Holczer, T., Schoch, E., Freudiger, J., Raya, M., Ma, Z., Kargl, F., Kung, A. and Hubaux, J.P. (2008a) Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine* **46**(11), 100–109.

Papadimitratos, P., Buttyan, L., Hubaux, J.P., Kargl, F., Kung, A. and Raya, M. (2007) Architecture for Secure and Private Vehicular Communications. *Proc. 7th International Conference on ITS Telecommunications ITST '07*, pp. 1–6.

Papadimitratos, P., Calandriello, G., Hubaux, J.P. and Lioy, A. (2008b) Impact of vehicular communications security on transportation safety. *INFOCOM Workshops 2008, IEEE*, pp. 1–6.

Papadimitratos, P., Gligor, V. and Hubaux, J.P. (2006a) Securing Vehicular Communications – Assumptions, Requirements, and Principles. *Workshop on Embedded Security in Cars (ESCAR)*, Berlin, Germany.

Papadimitratos, P., Kung, A., Hubaux, J.P. and Kargl, F. (2006b) Privacy and Identity Management for Vehicular Communication Systems: A Position Paper. *Workshop on Standards for Privacy in User-Centric Identity Management*, Zurich, Switzerland.

Papadimitratos, P., Mezzour, G. and Hubaux, J.P. (2008c) Certificate Revocation List Distribution in Vehicular Communication Systems (short paper). *ACM VANET*, San Francisco, CA.

Papadimitratos, P., Poturalski, M., Schaller, P., Lafourcade, P., Basin, D., Capkun, S. and Hubaux, J.P. (2008d) Secure neighborhood discovery: a fundamental element for mobile ad hoc networking. *IEEE Communications Magazine* **46**(2), 132–139.

Poturalksi, M., Papadimitratos, P. and Hubaux, J.P. (2008a) Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility. *ACM ASIACCS*, pp. 189–200, Tokyo, Japan.

Poturalksi, M., Papadimitratos, P. and Hubaux, J.P. (2008b) Towards Provable Secure Neighbor Discovery in Wireless Networks. *ACM Workshop on Formal Methods in Security Engineering*, Alexandria, VA, USA.

Rabin, M. (1989) Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. *Journal of the ACM*.

Raya, M., Papadimitratos, P., Aad, I., Jungels, D. and Hubaux, J.P. (2007) Eviction of Misbehaving and Faulty Nodes in Vehicular Networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Vehicular Networks*.

Raya, M., Papadimitratos, P., Gligor, V. and Hubaux, J.P. (2008) On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks. *IEEE INFOCOM*, Phoenix, Arizona, USA.

SeVeCom (2009) Secure Vehicular Communications. http://www.sevecom.org.

Shokrollahi, A. (2006) Raptor Codes. *IEEE/ACM Transactions on Networking*.

Syverson, P.F. and Stubblebine, S.G. (1999) Group Principals and the Formalization of Anonymity. *FM '99*, Toulouse, France.