Secure Vehicular Communication Systems: Design and Architecture

P. Papadimitratos^{+}, L. Buttyan†, T. Holczer†, E. Schoch*, J. Freudiger^{+}, M. Raya^{+},
Z. Ma*, F. Kargl*, A. Kung$^{\times}$, J.-P. Hubaux^{+}
$^{+}$School of Computer and Communication Sciences, EPFL, Switzerland
†Budapest University of Technology and Economics, Hungary
*Institute of Media Informatics, Ulm University, Germany
$^{\times}$Trialog, France

Abstract— **Significant developments have taken place over the past few years in the area of vehicular communication (VC) systems. Now, it is well understood in the community that security and protection of private user information are a prerequisite for the deployment of the technology. This is so, precisely because the benefits of VC systems, with the mission to enhance transportation safety and efficiency, are at stake. Without the integration of strong and practical security and privacy enhancing mechanisms, VC systems could be disrupted or disabled, even by relatively unsophisticated attackers. We address this problem within the *SeVeCom* project, having developed a security architecture that provides a comprehensive and practical solution. We present our results in a set of two papers in this issue. In this first one, we analyze threats and types of adversaries, we identify security and privacy requirements, and we present a spectrum of mechanisms to secure VC systems. We provide a solution that can be quickly adopted and deployed. In the second paper, we present our progress towards the implementation of our architecture and results on the performance of the secure VC system, along with a discussion of upcoming research challenges and our related current results.**

I. Introduction

After the deployment of various vehicular technologies, such as toll collection or active road-signs, vehicular communication (VC) systems have emerged. They comprise network nodes, that is, vehicles and road-side infrastructure units (RSUs), equipped with on-board sensory, processing, and wireless communication modules. Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication can enable a range of applications to enhance transportation safety and efficiency, as well as infotainment. For example, they can send warnings about 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.

The unique features of VC are a double-edged sword: a rich set of tools will be available, but a formidable set of abuses and attacks becomes possible. Consider, for example, an attacker that "contaminates" large portions of the vehicular network with false information: a single compromised vehicle can transmit false hazard warnings, which can then be taken up by all vehicles in both traffic streams; Or a tampered vehicle that forges messages to masquerade as an emergency vehicle to mislead other vehicles to slow down and yield; Or a different type of attacker that deploys a number of receivers, records messages transmitted by the vehicles, especially safety beacons that report the vehicle's location, to track a vehicle's location and transactions and to infer private information about its driver and passengers.

It is clear that to thwart such attacks, security and privacy-enhancing mechanisms are necessary; in fact, they are a prerequisite for deployment. Otherwise VC systems could make anti-social and criminal behavior easier, in ways that would actually jeopardize the benefits of their deployment. This has been recently well understood in academia, the industry, and among authorities; and a number of concerted efforts have been undertaken to design security architectures for VC systems.

A prominent example of these efforts is our three-year European-funded *Secure Vehicular Communications* (SeVeCom) Project (http://www.sevecom.org), which approaches its conclusion at the end of 2008. In this project, universities, car manufacturers, and car equipment suppliers collaborate on the design of a baseline architecture that provides a level of protection sought by users and legislators and is practical. Our baseline architecture is based on well-established and understood cryptographic primitives, but can also be tuned or augmented to meet more stringent future requirements.

In this paper, we first discuss the capabilities of attackers. Then, we present the requirements on which we base the development of our architecture. The basic aspects we seek to address are: identity and cryptographic key management, privacy protection, secure communication, and in-car protection. Next, we provide details on credential management and cryptographic support, which enable secure and privacy-enhancing communication. We conclude with a short discussion that ushers in our second article, which is concerned with implementation and performance issues, and upcoming research challenges.

II. Adversary Model

VC system entities can be *correct* or *benign*, that is, they may comply with the implemented protocols, or deviate from the protocol definition, i.e., be *faulty* or *adversarial*. Adversarial behavior can vary according to the implemented protocols and the capabilities of the adversary. Its incentive may be its own benefit or malice. Here, we do not consider benign faults, for example, communication errors, message delaying or loss, which can occur either under normal operational conditions or due to equipment failure. Instead, we focus on adversarial behavior that can cause a much larger set of faults. We do not describe attacks against individual VC protocols. Rather, we survey the capabilities of adversaries and discuss aspects relevant to the VC context. A more detailed exposition, which also discusses models used in other types of distributed systems, is available in [9].

Even though the VC protocol implementations will be proprietary, open definitions of standards can provide attackers with detailed knowledge about the system operation. Any wireless device that runs a rogue version of the VC protocol stack poses a threat. Attackers can either be passive or active.

Active adversaries can meaningfully *modify* in-transit messages they relay, beyond the modifications the protocol definitions allow or require them to perform. Or, more generally, they can *forge*, i.e., synthesize in a manner non-compliant to the protocols and system operation, and *inject* messages. As adversaries are aware of the VC protocols, they can choose any combination of these actions according to their own prior observations (messages they received) and the protocol they attempt to compromise. An active adversary may also *jam* communications, by interfering deliberately to prevent other devices within its range to communicate. Or, it can *replay* messages that it received and were previously transmitted by other system entities. In contrast to active adversaries, *passive attackers* only learn information about system entities and cannot affect or change their behavior.

It is important to distinguish adversaries equipped with cryptographic keys and credentials that entitle them to participate in the execution of the VC system protocols. We denote those as *internal* adversaries. In contrast, adversaries that do not possess such keys and credentials are *external*. We emphasize that the possession of credentials does not guarantee correct operation of the nodes. For example, the *on-board units* (OBUs) can be tampered with and their functionality modified (e.g., by installing a rogue version of the protocol stack). Or, the cryptographic keys of an RSU or a vehicle can be compromised (e.g., physically extracted from an unattended vehicle) and be utilized by an adversarial device. If this were the case, a node with multiple (compromised) keys could appear as multiple nodes.

More generally, *multiple adversarial nodes* can be present in the network at different locations. They can be acting *independently* or they may *collude*, i.e., exchange information and coordinate their actions, in order to mount a more effective attack.[1] For example, they could all report an imaginary event (e.g., traffic jam or accident), in order to mislead correct nodes into thinking this is indeed the case. Over time, the set of adversarial nodes can change both in numbers and locations. On the one hand, the compromised nodes, for example, illegally modified vehicles, can increase over time, as drivers may have some benefit in doing so. On the other hand, fault detection mechanisms and diagnostics, along with policy enforcement can lead to gradual eradication of faulty devices.

Overall, however, it is reasonable to expect that only a *relatively small fraction* of the VC devices would be adversaries. Of course this depends on the appropriate design of the system, which should not allow for easy exploits (e.g., malware propagation). Moreover, the majority of the users do not have the expertise and the motivation to tamper with their VC devices and maintenance can address the majority of equipment faults.

Given a small fraction of faulty (adversarial) devices, the adversary should have an overall *limited physical presence*. As the transmission range of faulty devices cannot be unbounded, even if they had customized hardware that exceeds the communication range of vehicular or road-side devices, the adversaries can affect only a fraction of the VC system area. Within this area, they can cause denial of service and do so in a selective manner, i.e., erase one or more messages sent by other nodes. This does not preclude that a few adversarial devices surround a correct node (vehicle) at some point in time. But most often and in most locations, correct nodes should encounter few or only a single adversary.

Due to the nature of VC systems, with vehicles equipped with a number of sensors, an exchange of false measurements can compromise the VC-enabled applications. An arguably convenient attack, in the sense that it may be relatively easy to mount, would be to *control the sensory inputs* to the OBU, instead of attempting to compromise the OBU or its cryptographic keys. Tampering with a sensor or with the OBU-sensor connection may indeed be simpler. It is not easy to classify an *input-controlling adversary* as external or internal. On the one hand, no access to credentials and cryptographic material is necessary. On the other hand, messages generated and transmitted due to the input-controlling adversary originate from a legitimate system participant. What we should note though is that such an adversary is relatively 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.

III. Security Requirements

The problem at hand is to secure the operation of VC systems, by designing protocols that mitigate attacks and thwart deviations from the implemented protocols to the greatest possible extent. Different protocols have their own specifications. Rather than providing an exhaustive enumeration of sought properties per protocol and application, first we identify a set of stand-alone requirements. Then, we outline a number of example VC applications along with the related security requirements.

The identified stand-alone security requirements are the following:

[1] We emphasize though that even in that case, adversaries are *computationally limited* and unable to break keys of other nodes.

Message Authentication and Integrity, to protect against any alteration and allow the receiver of a message to corroborate the sender of the message.

Message Non-Repudiation, so that the sender of a message cannot deny having sent a message.

Entity Authentication, so that a receiver is ensured that the sender generated a message *and* has evidence of the *liveness* of the sender. In other words, ascertain that 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, to determine via specific system-wide policies the assignment of distinct roles to different types of nodes and their allowed actions within the system. 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, to keep the content of a message secret from those nodes not authorized to access it.

Accountability, to be able to map security-related events to system entities.

Privacy Protection, to safeguard private information of the VC system users. This is a general requirement that relates to the protection of private information stored off-line. In the context of communication, which is the object of SeVeCom, we are interested in **anonymity** for the actions (messages and transactions) of the vehicles. We elaborate on the VC-specific aspects that we seek to address next.

For privacy, along with security, we focus on private vehicles (e.g., excluding emergency vehicles, buses, etc) because the operation of all other VC nodes, including RSUs, does not raise any privacy concerns, and all the other nodes should be readily identifiable. A primary concern for VC systems is to provide *location privacy*, by preventing others (any *observer*) from learning past or future locations of a VC system user (vehicle driver or passenger). With our focus on VC, we can safeguard location privacy by seeking to satisfy a more general requirement, anonymity for the vehicle message transmissions.

Ideally, it should be impossible for any observer to learn if a specific vehicle has transmitted or will transmit a message (more generally, take an action according to a VC protocol), and it should be impossible to link any two or more messages (in general, actions) of the same vehicle. Even if an observer tried to guess, there should be only a low probability of linking a vehicle's actions or identifying it among the set of all vehicles, the *anonymity set*. We will elaborate on this notion when we discuss below the management of identities and credentials for VC system entities.

Rather than aiming for this strong anonymity, we require a relatively weaker level of protection: 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 vehicle over a protocol-selectable period of time τ can always be linked by an observer that receives them. But messages m_1, m_2 generated at times t_1, t_2 such that $t_2 > t_1 + \tau$ cannot. In terms of the observer, we assume that its physical presence is bounded, as stated earlier for the adversary.

	Feature			Requirement		
Application	**Safety Appl.**	**V2V / V2I**	**Multi-Hop**	**Authentication**	**Integrity**	**Privacy**
Intersection collision warning	√	V2V		2	2	2
Emergency vehicle signal	√	V2I	√	2	2	0
Work zone warning	√	V2I	√	1	2	0
Forward collision warning	√	V2V	√	2	2	2
Cooperative adaptive cruise control		V2V	√	2	2	2

TABLE I

SAMPLED VC APPLICATIONS: FEATURES AND IMPORTANCE OF SECURITY REQUIREMENTS.

In addition, features that enhance **availability** are required, to enable protocols and services to remain operational even in the presence of faults, malicious or benign. This implies resilience to resource depletion attacks, as well as self-stable protocols that resume their normal operation after the "removal" of the faulty participants.

Based on these considerations, SeVeCom performed a detailed requirements analysis where general application characteristics and security requirements were assessed for a large number of VC applications [6]. Table I shows a small excerpt from this analysis, with higher values indicating a high importance of a given requirement. For example, for a work zone warning message, it may be relatively low in importance to rigidly determine its recency. For a collision avoidance application though, it is crucial to ensure the message recency. Of course, for both applications, it is critical to ensure that no message content is fabricated by an attacker. Regarding privacy protection, this is not required for infrastructure- or public vehicle-sent messages such as the work zone and emergency vehicle warnings.

IV. SECURE VC SYSTEM OVERVIEW

Our architecture addresses the following fundamental issues: (i) *identity, credential, and key management*, and (ii) *secure communication*. Seeking to satisfy the requirements we outlined earlier in this article, we focus primarily on securing the operation of the wireless part of the VC system, and enhancing the privacy of its users. We are fully aware of the projected co-existence of VC-specific and TCP/IP protocol stacks in VC systems. Moreover, towards further strengthening our architecture, we have investigated and developed approaches to address *in-car protection* and *data consistency*, discussed in [7]. An abstract view of the secure VC system, with *nodes* (vehicles and RSUs) and *authorities* (CA_A and CA_B), is shown in Fig. 1. We outline next the main elements of our architecture.

Authorities Drawing from the analogy with existing administrative processes and automotive authorities (e.g., city or state transit authorities), we assume that a large number of *Certification Authorities* (CAs) will be instantiated. Each CA is responsible for a *region* (national territory, district, county, etc.) and manages identities and credentials of all nodes registered with it. To enable interactions between nodes from different regions, CAs provide certificates for other CAs (cross-certification) or provide *foreigner certificates* to vehicles that are registered with another CA when they cross the geographical boundaries of their region [10].

Node Identification Each node is registered with only one CA, and has a unique *long-term* identity and a pair of *private* and *public* cryptographic keys, and 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 CA is also responsible for the *eviction* of nodes or the *withdrawal* of compromised cryptographic keys via the revocation of the corresponding certificates. In all cases, the interaction of nodes with the CA is infrequent and intermittent, with the road-side infrastructure acting as a gateway to and from the vehicular part of the network, with the use of other infrastructure (e.g., cellular) being also possible. The conceptual view of VC nodes is illustrated in Fig. 2. The node identity and credential management and the role of the HSM, methods to secure V2V and V2I communication, and CA-vehicle interactions (V2CA) that include the issuance of *short-term credentials* to secure vehicle transmissions, are discussed in the rest of the paper. The in-car system and data processing functionality are discussed in [7].

Hardware Security Module (HSM) We envision that both vehicles and RSUs are equipped with an HSM, whose purpose is to store and physically protect sensitive information and provide a secure time base. This information is primarily private keys for signature generation. If modules were to be tampered with, to extract private keys, the physical protection of the unit would ensure that the sensitive information (private keys) would be erased, thus preventing the adversary from obtaining them. In addition, the HSM performs all private key cryptographic operations with the stored keys, in order to ensure that sensitive information never leaves the physically secured HSM environment. Essentially, the HSM is the basis of trust; without it, private keys could be compromised and their holders could masquerade as legitimate system nodes.

Secure Communication Digital signatures are the basic tools to secure communications and are used for all messages. To satisfy both the security and anonymity requirements, we rely on a *pseudonymous authentication* approach. Rather than utilizing the same long-term public and private key for securing communications, each vehicle utilizes multiple *short-term* private-public key pairs and certificates. A mapping between the short-term credentials and the long-term identity of each node is maintained by the CA.

The basic idea is that (i) each vehicle is equipped with multiple certified public keys (pseudonyms) that do not reveal the node identity, and (ii) the vehicle uses each of them for a short period of time, and then switches to another, not previously used pseudonym. This way, messages signed under different pseudonyms cannot be linked. Signatures, calculated over the message payload, a time-stamp and the coordinates of the sender, can be generated by the originator of a message, as well as relaying nodes, depending on the protocol functionality. We provide security for frequently broadcasted *safety beacon* messages, *restricted flooding* of messages within a geographical region or a hop-distance from the sender, and *position-based routing* used to transmit messages through a single route of relay nodes, where the nodes select as next hop their neighbor with minimum remaining geographical distance to the destination position.

V. Credential Management and Cryptographic Support

The management of credentials, both short and long-term, is undertaken by the CAs that are also responsible for the revocation of credentials for any node if needed, as well as holding the node accountable by mapping node communications to its long-term identity. Public key operations are performed by the OBU, but all private key operations are performed by the HSM, which is essentially the trusted computing base of the secure VC system.

A. Identity and Credential Management

1) Long-Term Identification: Each node X has a unique long-term identity ID_X, which will be the outcome of an agreement between car manufacturers and authorities, similar to the use of Vehicle Identification Numbers (VINs). Identifiers of the same format will be assigned both to vehicles and road-side units. Each identity is associated with a cryptographic key pair (SK_X, PK_X), and a set of attributes of node X. The attributes reflect technical characteristics of the node equipment (for example, type, dimensions, sensors and computing platform), as well as the role of the node in the system. Nodes can be, for example, private or public vehicles (buses), or vehicles with special characteristics (police patrol cars), or RSUs, with or without any special characteristics (offering connectivity to the Internet). The assignment of an identity, the selection of attributes appropriate for each node, and the generation of the certificate are performed "off-line," at the time the node is registered with the CA. The lifetime of the certificate should be long, following the node life-cycle (or a significant fraction of it).

2) Short-Term Identification: To obtain pseudonyms, a vehicle V's HSM generates a set of key pairs $\{(SK_V^1, PK_V^1), ..., (SK_V^i, PK_V^i)\}$ and sends the public keys to a corresponding CA via a secured communication channel. V utilizes its long-term identity ID_V to authenticate itself to the CA. The CA signs each of the public keys, PK_V^i, and generates a set of pseudonyms for V. Each pseudonym contains an identifier of the CA, the lifetime of the pseudonym, the public key, and the signature of the CA; thus, no information about the identity of the vehicle.

Pseudonyms are stored and managed in the on-board pseudonym pool, with their corresponding secret keys kept in the HSM. This ensures that each vehicle has exactly one key pair (own pseudonym pseudonym and private key) that is active during each time period. Moreover, once the switch from the (SK_j, PK_j) to the $j+1$-st key pair (SK_{j+1}, PK_{j+1}) is made, no messages can be further signed with SK_j; even if the certificate for PK_j is not yet expired. In other words, pseudonymity cannot be abused: For example, a rogue vehicle cannot sign multiple beacons, each with a different SK_j over a short period, and thus cannot appear as multiple vehicles.[2]

A vehicle needs to contact the CA, infrequently but regularly, to obtain a new set of pseudonyms. For example, if a vehicle utilizes pseudonyms in set i, it obtains the $(i+1)$-st set of pseudonyms while it can still operate with the i-th set. It switches to the $(i+1)$-st set once no pseudonym in the i-th set can be used. We term this process a *pseudonym refill*.

Due to the requirement for accountability, the CA archives the issued pseudonyms together with the vehicle's long-term identity. In case of an investigation, an authorized party can ask the CA to perform a *pseudonym resolution*: reveal the link of a specific pseudonym to the long-term identity of the vehicle (this leading further to its registered owner).

By using the same pseudonym only for a short period of time and then switching to a new one, vehicle activities can be linked only during the period when the same pseudonym is used. Changing pseudonyms makes it difficult for an adversary to link messages from the same vehicle and track its movements. However, the inclusion of the identity of the CA_A issuing the credential (pseudonym) implies that the vehicle is part of the set of all vehicles registered with CA_A. In fact, this is the anonymity set of vehicle V. This implies that, for example, a Swiss vehicle should be anonymous within the set of all Swiss vehicles.

This division of vehicles into disjoint subsets, one per CA, allows an observer to rule out a significant portion of vehicles given geographical constraints. Consider again a Swiss vehicle, driving in the East Balkans where it is not likely to encounter numerous other vehicles with the same registration. An observer could correctly guess with high probability that all Swiss pseudonyms (and thus associated messages) are used by the same Swiss vehicle. To prevent such inferences, we require that vehicles crossing the boundaries of a foreign region, B, obtain short-term credentials from the local CA_B [10]. In our example, V would have to first prove to CA_B it is registered with CA_A, then obtain pseudonyms by CA_B, and use them exclusively while in region B. This way, it would avoid "standing out" in region B, appearing to any observer of the VC system traffic as part of the anonymity set B.

B. Hardware Security Module

The Hardware Security Module (HSM) is the trusted computing base of the SeVeCom security architecture. It stores the private cryptographic key material, and provides cryptographic functions to be used by other modules. The HSM is physically separated from the On-Board Unit (OBU), and it has some tamper resistant properties in order to protect the private key material against physical attacks. The HSM consists of a CPU, some non-volatile memory, a built-in clock, and some I/O interface. In addition, the HSM has a built-in battery in order to power the clock and the tamper detection and reaction circuitry.

The main HSM functions include cryptographic operations, as well as key and device management functions. The main cryptographic operations provided by the HSM are the digital signature generation and the decryption of encrypted messages. The digital signature generation function is mainly used by the secure communication module (see Sec. VI) for signing outgoing messages. The HSM always includes a timestamp in every signature that it generates, which makes it possible to detect replay attacks. The decryption function is mainly used by the pseudonym handling application, which receives the anonymous certificates in an encrypted form from the pseudonym provider.

The HSM handles short-term keys for the short-term identification and long-term keys for the long-term identification of the vehicle. These keys are generated by the HSM, and only the public keys are output from the device. The generation of short-term keys can be initiated by any application running on the OBU. In contrast, the long-term keys are generated at manufacturing time, however, they can be updated later by trusted authorities.

Device management and long-term key updates are achieved through signed commands from the CA. In order to verify the signature on these commands, the HSM stores trusted root public keys that are loaded into the device during the initialization procedure in a secure environment. We envision two such *root public keys*, K_1 and K_2, in the HSM, with the corresponding private keys held by the CA. In case one of the CA'a private keys is compromised, the corresponding public key, say K_1, can be revoked, as discussed in the next paragraph. The revocation command must be signed with the private key corresponding to K_1 itself. Once K_1 is revoked, a new key K_1' can be loaded into the HSM by a command signed with the private key corresponding to K_2. In addition, when K_1 is revoked, the HSM does not accept commands aimed at revoking K_2. This scheme ensures secure root key update unless both root keys are compromised.

As discussed next, CA commands can include revocation of the entire device. The revocation of the HSM is achieved by a signed kill command that deletes every piece of information from the memory, making the device unusable. Further device management functions include device initialization, and clock synchronization. During device initialization, the main parameters of the HSM, as well as the root public keys are loaded in the HSM. Clock synchronization allows for synchronizing the internal clock of the HSM to a trusted external clock.

[2] The CA could prevent abuse of the pseudonymity by issuing short-term certificates with non-overlapping lifetimes. We also note that multiple pseudonyms can be active simultaneously only when used for completely disjoint communication (e.g., one for all safety messaging and one for infotainment downloads).

C. Revocation

The certificates of faulty nodes have to be revoked, to prevent them from causing damage to the VC system. Revocation can be decided by the CA because of administrative or technical reasons. The basic mechanism to achieve this is *Certificate Revocation Lists* (CRLs) the CA creates and authenticates. The challenge is to distribute effectively and efficiently the CRLs, which can be achieved by a combination of methods illustrated in Fig. 3.

We leverage on the road-side infrastructure to distribute CRLs. We find that with RSUs placed on the average some kilometers apart, and with CRL distribution by each RSU at a few kbps, all vehicles can obtain CRLs of hundreds of kilobytes over a time period of an average commute [10]. This is achieved primarily due the use of encoding of CRLs into numerous (cryptographically) self-verifiable pieces and low-rate broadcast transmission of CRL pieces. In areas with no RSUs, the V2V CRL distribution initiated by vehicles that were previously in contact with RSUs, or use of other communication technologies, could have a complementary role. The size of CRLs and the overall amount of revocation information to be distributed can still be a challenge. At first, the collaboration between CAs, so that CRLs contain only regional revocation information, can keep the CRL size low [10].

Revocation can leverage on the HSM, with the CA initiating the RHSM (Revocation of the HSM) protocol [13], issuing a “kill” command signed with the private key corresponding to one of the root public keys. If a HSM receives a kill command, it deletes everything from its memory including its own private keys, to prevent the generation of any new keys or signatures by the compromised module. The CA determines the location of the vehicle and sends the kill command via the nearest RSU(s). The HSM has to confirm the reception of this command by sending an ACK before erasing the long term signature generation key (SK_X). If communication via the RSUs fails (i.e., an ACK is not received after a timeout), the CA can broadcast the command via the RDS (Radio Data System).

If the adversary controls the CA-HSM communication, the CRL-based revocation has to be performed. This can also be done via the RC^{2}RL (Revocation using Compressed Certificate Revocation Lists) protocol [13], which can reduce the size of CRLs by a lossy compression scheme, notably Bloom filters, to the extent they could be transmitted even over the RDS. The identification of a revoked certificate in the Bloom filter is always possible (zero false negative rate), along with a configurable low false positive rate. An occasional revocation of “innocent” credentials, traded-off for compression (efficiency), is not an issue when RC^{2}RL revokes large numbers of short-term credentials.

The inclusion of credentials in a CRL implies that the CA has established the need to revoke the node. If this is because of faulty behavior, the absence of an omnipresent monitoring facility makes the detection harder. Moreover, CRLs will be issued rather infrequently (e.g., once per day), thus leaving a vulnerability window until a faulty node is revoked. To address this, we propose that misbehavior detection is left to vehicles, which can then defend themselves by locally voting off and excluding misbehaving vehicles. We propose the use of two localized defense schemes, MDS (Misbehavior Detection System) and LEAVE (Local Eviction of Attackers by Voting Evaluators) [13]. The first allows the neighbors of a misbehaving node detect it, and the second enables them to exclude it from the local VC operation. After a LEAVE execution, the evaluators report the misbehaving node to the CA; a node can be revoked by the CA, using one of the previously described approaches, after having been evicted a threshold number of times by its (changing) neighbors.

VI. Secure Communication

A. Secure Beaconing

Beaconing denotes periodic single-hop broadcasts typically used for the so-called Cooperative Awareness applications. In order to create awareness of other vehicles in the vicinity, every beacon contains information on the sender’s status such as vehicle position, speed, and heading. The frequency of beacon packets is expected to range from about 10Hz to 1Hz for most use cases.

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 MAC layer for transmission, the sending node (V) calculates a signature $sig(m)$, using the private key $SK(j, V)$ corresponding to the j-th pseudonym $PK(j, V)$ that is currently in use. V includes a time-stamp and a geographic position at the instant of transmission, together they are termed a geo-stamp. Beyond the signature $sig(m)$ that covers all these fields, V also attaches $Cert_A\{PK(j, V)\}$, which attests the validity of $SK(j, V)$. The receiver can verify message signatures using $PK(j, V)$ in the attached certificate, $Cert_X\{PK(j, V)\}$, which can be verified using the pre-installed public key of CA_A (See Fig. 4).

These measures achieve four goals. First, the receiver of a beacon message can verify that its sender is a valid participant of the VC system (either vehicle or RSU). Second, no node can impersonate another node without compromising its HSM. Third, the integrity of the message is protected, as manipulations are detected if the signature is invalid. Finally, the use of the geo-stamp, along with the signature, allows for the detection of replay attacks. Details on replay protection mechanisms follow in the next section.

B. Secure Neighbor Discovery

Cooperative awareness or safety messaging allow vehicles to discover a frequently updated view of other vehicles in proximity, called *physical neighbors*. In addition, for the purpose of communication, it is important that vehicles also discover other nodes (vehicles or RSUs) that are directly reachable, that is, called *communication neighbors*. Typically, it is assumed that if two nodes are communication neighbors, then they are physical neighbors, and vice versa. However, this is not the case, because adversaries mount *relay attacks*, receiving and quickly retransmitting (replaying) messages of remote nodes [11].

The inclusion of sender time-stamp and location, along with authentication, enable our system to perform provably secure neighbor discovery against *external* adversaries [12]. The basic idea is to estimate the sender-receiver distance based on own coordinates and the location in the received message and time-of-flight (difference between own time and received timestamp). For a protocol-selectable acceptable neighbor range, the receiving node accepts the sender as communication neighbor when the two distance estimates are equal and the sender is authenticated. As a result, vehicles can be ensured that their *neighbor table* includes only nodes that are indeed communication neighbors.

C. Secure GeoCast

The range covered by one-hop beaconing is often not sufficient and information on events such as accidents needs to be disseminated in relatively large areas. This is achieved by Geocast, which comprises three elements: (i) addressing of a geographically defined destination region, (ii) forwarding towards this region and (iii) distribution of the packet within the destination region. Position-based routing, that is, multi-hop, single-path forwarding of packets towards a geographically defined destination, has been shown to be well-suited to the dynamics of vehicular networks. Position-based routing is realized by greedy routing protocols such as GPSR or CGGC. The distribution of messages among all nodes within the destination region can be done by simple flooding or by more efficient approaches to multi-hop broadcast. In the case of simple flooding, every node inside the destination region rebroadcasts the message once and records its sequence number to suppress re-broadcasting of the same message.

As a basic security measure for both position-based routing and message distribution, source nodes sign created messages and then attach the corresponding certificate, similarly to the functionality for Secure Beaconing. Moreover, forwarding nodes can also sign packets they relay, so that they can be authenticated by the next-hop relay [5]. 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 section.

As beaconing is the basis for position-based forwarding decisions, the location given in beacons can be forged, with data delivery failures (when traffic is attracted by the attacker), and increased network load (due to routing loops). We propose a position verification approach, based on plausibility heuristics, which is capable of detecting such position falsifications [8]. Second, changing pseudonyms for privacy reasons leads to increased instability in nodes' neighbor tables. This can result in transmission faults to the next hop, because a node is not reachable any more after a pseudonym change, which deteriorates routing performance [14]. To balance network and privacy needs, we can extend the routing mechanisms by a MAC layer callback that notifies the routing layer about missed neighbors. Finally, to mitigate resource depletion attacks, with an internal adversary distributing at high rates messages across a large destination region, we propose rate limiting.

D. Pseudonym Handling

An adversary analyzing which certificates are attached to signed messages can track the location of vehicles over time. Hence, we propose to load each vehicle with multiple certified public keys (i.e., pseudonyms) that it uses for short periods of time. If pseudonyms are changed at appropriate times and locations, messages signed under different pseudonyms are hard to be linked by an adversary.

As the adversary could use information from other layers of the communication stack to track vehicles (e.g., MAC, IP), a change of pseudonym should be accompanied by a change of the vehicle identifiers in underlying protocols as well. Still, using the location contained in messages to match pseudonyms, an adversary can indirectly identify vehicles by predicting the next position of vehicle, even if a vehicle has a new pseudonym. Cloaking of location information [4] is not a solution as it would jeopardize the use of safety applications. In [2], we propose that vehicles change pseudonyms in regions not monitored by an adversary. These regions are called mix zones [1] because by changing pseudonyms the vehicles mix with each other. We also suggest that vehicles change their pseudonym at regular intervals to maximize the probability of changing pseudonyms in a mix zone. In [3], we explore another approach that creates un-monitored regions by encrypting communications (i.e., cryptographic mix zones) in small regions with the help of the road infrastructure.

The general idea of mix zones is explained as follows: If only one vehicle changes its pseudonym in a mix zone, an adversary observing vehicles entering and exiting the region would trivially track vehicles because only one pseudonym has changed. But if more than one vehicle change their pseudonyms in a mix zone, the adversary should consider every possible match for entering and exiting vehicles and estimate the most likely matches given its belief about the mobility of vehicles, the time to traverse the mix zone and the geometry of the mix zone (Fig. 5). An adversary would thus find several possible matches weighted with different probabilities. To measure the amount of location privacy achieved by vehicles in the mix zone, we capture the uncertainty of the adversary with the notion of entropy as defined in [15]. With this metric, if all possible matches look equally likely to an adversary, (i.e. the mix zone is very unpredictable), the adversary would be highly inefficient in tracking vehicles. In general, the more vehicles there are in a mix zone, the more difficult it is for an adversary to obtain a good estimation of probabilities as many combinations are possible. If the mobility of the vehicles is such that they are equally likely to enter/exit the mix zone from any road, it would be difficult for the adversary to obtain precise predictions.

When vehicles change pseudonyms in un-monitored regions of the network, then mix zones are large and it is difficult for an adversary to obtain good estimations. However, when mix zones are created by the use of cryptography, then they tend to be smaller, and thus must be located appropriately to maximize their effectiveness (e.g., at traffic intersections). Hence, linking messages signed under different pseudonyms becomes increasingly hard over time and space for an adversary. As vehicles change pseudonyms several times before reaching their destinations, the adversary will accumulate more uncertainty similarly to a mix network [15], mobile nodes can achieve a high level of location privacy.

VII. Conclusions

We have developed a security architecture for VC systems, aiming at a solution that is both comprehensive and practical. We have studied systematically the problem at hand, identifying threats and models of adversarial behavior, as well as security and privacy requirements that are relevant to the VC context. We have introduced a range of mechanisms, to handle identity and credential management, and to secure communication while enhancing privacy. In the second paper of this contribution, we discuss implementation and performance aspects, present a gamut of research investigations and results towards further strengthening secure VC systems and addressing the remaining research challenges.

Authors

PANAGIOTIS (PANOS) PAPADIMITRATOS (panos.papadimitratos@epfl.ch) received his Ph.D. degree from Cornell University in 2005. After a research associate position at Virginia Tech, he is currently a senior researcher at EPFL. His research is concerned with security, networking protocols, and wireless and mobile systems. He has authored more than 50 technical publications on these topics, delivered several tutorials including one at ACM Mobicom 07, and has served on the program committees of ACM Mobihoc, WiSec, ASIACCS, and VANET, and IEEE Infocom, among other venues. His Web page is http://people.epfl.ch/panos.papadimitratos.

LEVENTE BUTTYAN (buttyan@crysys.hu) received the M.Sc. degree in Computer Science from the Budapest University of Technology and Economics (BME) in 1995, and the Ph.D. degree from the Swiss Federal Institute of Technology, Lausanne (EPFL) in 2002. In 2003, he joined the Department of Telecommunications at BME, where he currently holds a position as Associate Professor and works in the Laboratory of Cryptography and Systems Security (CrySyS). His research interests are in the design and analysis of security protocols for wireless networks, including wireless sensor networks, mesh networks, vehicule communication systems, and RFID systems. More information is available at http://www.hit.bme.hu/ buttyan/

TAMÁS HOLCZER received his M.Sc. degree in Computer Science from the Budapest University of Technology and Economics (BME) in 2005. Since 2005 he has been working as a PhD student in the Laboratory of Cryptography and System Security (CrySyS) under the supervision of Levente Buttyán at BME. His research interests are in the area of privacy problems of ad hoc networks especially location privacy in vehicular networks. http://www.crysys.hu/members/tholczer/

ELMAR SCHOCH (elmar.schoch@uni-ulm.de) received his degree in Computer Science from Ulm University in 2005, working on security of vehicular networks at DaimlerChrysler Telematics research. Afterwards, he joined Ulm University as a researcher in the SEVECOM project, continuing to work on inter-vehicle communication with a special focus on robust and secure communication mechanisms.

JULIEN FREUDIGER received his M.Sc. degree in Communication Systems in 2006 from EPFL, Switzerland. He completed his master thesis at Qualcomm Inc., CA, where he was involved in the deployment of VoIP on CDMA networks. He is currently a Ph.D. student at EPFL under the supervision of Prof. Jean-Pierre Hubaux. His research focuses on the emergence of privacy in mobile wireless networks. http://people.epfl.ch/julien.freudiger

MAXIM RAYA received his B.Eng. degree in Computer and Communications Engineering in 2002 from the American University of Beirut, Lebanon. He is currently pursuing his Ph.D. studies under the supervision of prof. Jean-Pierre Hubaux at EPFL. His research interests are in the area of security in wireless networks, and especially vehicular networks. He served in the program committees of VANET 2007 and VANET 2008. http://people.epfl.ch/maxim.raya.

ZHENDONG MA (zhendong.ma@uni-ulm.de) received his MSc in Telecommunications in 2005 from Technical University of Denmark. At present he is a Ph.D. candidate at Institute of Media Informatics, Ulm University, Germany. His research interests include privacy in vehicular communications, security in communication systems, and mobile and ubiquitous computing.

FRANK KARGL (frank.kargl@uni-ulm.de) received his doctorate degree in 2003 working on security in mobile adhoc networks. His research interest center around mobile and dynamic systems, especially inter-vehicular networks with a focus on their security and privacy. Today he is leading a research team that is addressing those challenges and is involved in various national and European research projects like SeVeCom and PRECIOSA.

ANTONIO KUNG (antonio.kung@trialog.com) has more than 25 years experience on embedded systems. He cofounded Trialog in 1987 where he is in charge of the development of software products such as protocols for the EHS/KNX home systems network, real-time kernels, and Java technology. He has led a number of security projects (e-PASTA, the security part of GST, Sevecom). He is currently co-chairing the eSecurity WG of the eSafety forum which focuses in particular on data protection. He holds a Master's degree from Harvard University, USA and an engineering degree from Ecole Centrale Paris, France.

JEAN-PIERRE HUBAUX (jean-pierre.hubaux@epfl.ch) joined the faculty of EPFL in 1990. His research activity is focused on wireless networks, with a special interest in security and cooperation issues. He has recently completed a graduate textbook, Security and Cooperation in Wireless Networks (http://secowinet.epfl.ch), with Levente Buttyan. He is chairman of the steering committees of ACM Mobihoc and ACM WiSec, and a member of the Swiss Federal Communications Commission (ComCom), the ”Swiss FCC.” For more information, check http://people.epfl.ch/jean-pierre.hubaux.

REFERENCES

[1] A. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46– 55, Jan.-Mar. 2003.

[2] L. Buttyan, T. Holczer, and I. Vajda. On the effectiveness of changing pseudonyms to provide location privacy in vanets. In *Proceedings of the Third European Workshop on Security and Privacy in Ad hoc and Sensor Networks*, volume 4572, pages 129–141, July 2007.

[3] J. Freudiger, M. Raya, M. Felegyhazi, P. Papadimitratos, and J.-P-Hubaux. Mix-zones for location privacy in vehicular networks. In *Proceedings of the First International Workshop on Wireless Networking for Intelligent Transportation Systems (Win-ITS)*, Vancouver, British Columbia, Canada, August 2007.

[4] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the ACM MobiSys*, pages 31 – 42, New York, NY, USA, 2003.

[5] C. Harsch, A. Festag, and P. Papadimitratos. Secure Position-Based Routing for VANETs. In *Proceedings of the IEEE 66th Vehicular Technology Conference (VTC2007-Fall)*, pages 26–30, October 2007.

[6] F. Kargl, Z. Ma, and E. Schoch. Security engineering for vanets. In *Proceedings of the fourth Workshop on Embedded Security in Cars (ESCAR)*, pages 15–22, Berlin, Germany, November 2006.

[7] F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, B. Wiedersheim, E. Schoch, T.-V. Thong, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux. Secure vehicular communications: Implementation, performance, and research challenges. *IEEE Communcations Magazine*, November 2008.

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

[9] P. Papadimitratos, V. Gligor, and J.-P. Hubaux. Securing vehicular communications - assumptions, requirements, and principles. In *4th Workshop on Embedded Security in Cars (ESCAR)*, pages 5–14, Berlin, Germany, 2006.

[10] P. Papadimitratos, G. Mezzour, and J.-P. Hubaux. Certificate Revocation List Distribution in Vehicular Communication Systems (short paper). In *ACM VANET*, San Francisco, CA, USA, September 2008.

[11] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. Čapkun, and J.-P. Hubaux. Secure Neighborhood Discovery: A Fundamental Element for Mobile Ad Hoc Networking. *IEEE Communications Magazine*, 46:132–139, February February 2008.

[12] M. Poturalksi, P. Papadimitratos, and J.-P. Hubaux. Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility. In *ACM ASIACCS*, pages 189–200, Tokyo, Japan, March 2008.

[13] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux. Eviction of Misbehaving and Faulty Nodes in Vehicular Networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Vehicular Networks*, 25(8):1557–1568, October 2007.

[14] E. Schoch, F. Kargl, T. Leinmüller, S. Schlott, and P. Papadimitratos. Impact of Pseudonym Changes on Geographic Ad Hoc Routing. In *Proceedings of The Third European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS)*, volume 4357, pages 43–57, Hamburg, Germany, September 2006.

[15] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In Paul Dingledine, Roger; Syverson, editor, *PET*, pages 259–263, August 2002.

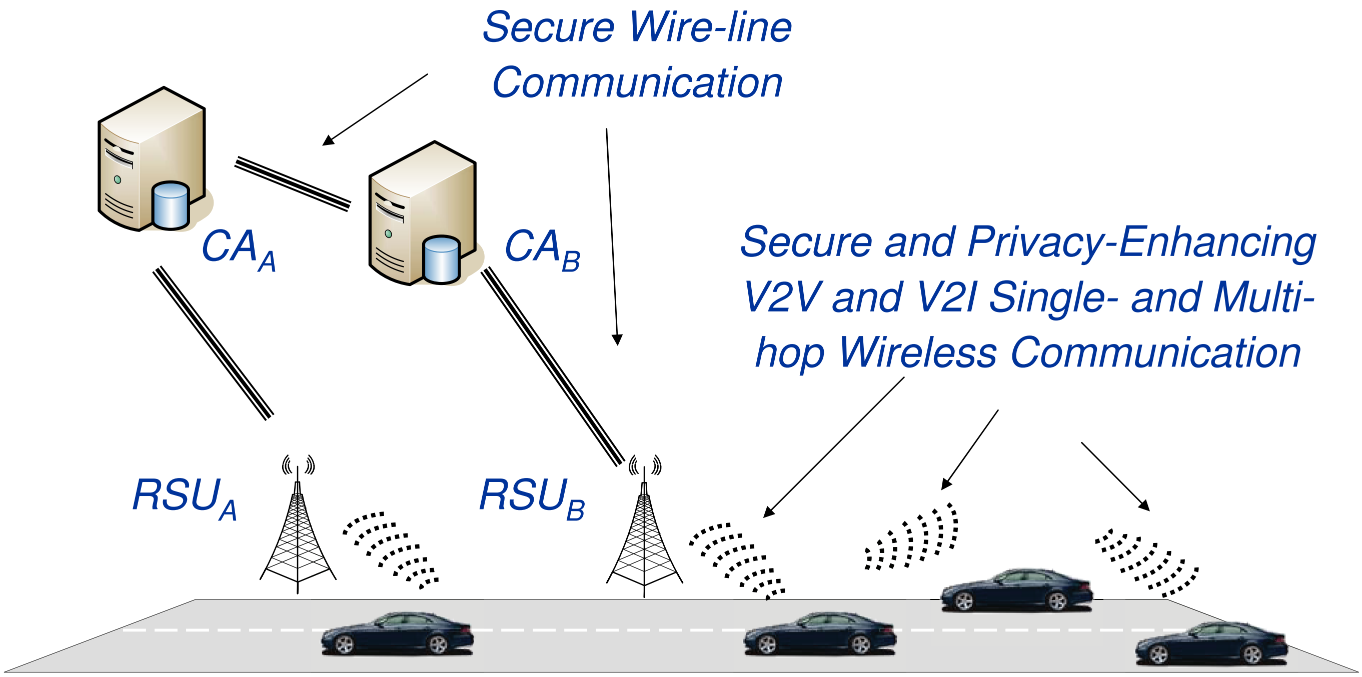
Secure Wire-line Communication
CA$_A$
CA$_B$
Secure and Privacy-Enhancing V2V and V2I Single- and Multi-hop Wireless Communication
RSU$_A$
RSU$_B$

Fig. 1. Abstract View of the Secure Vehicular Communication System.

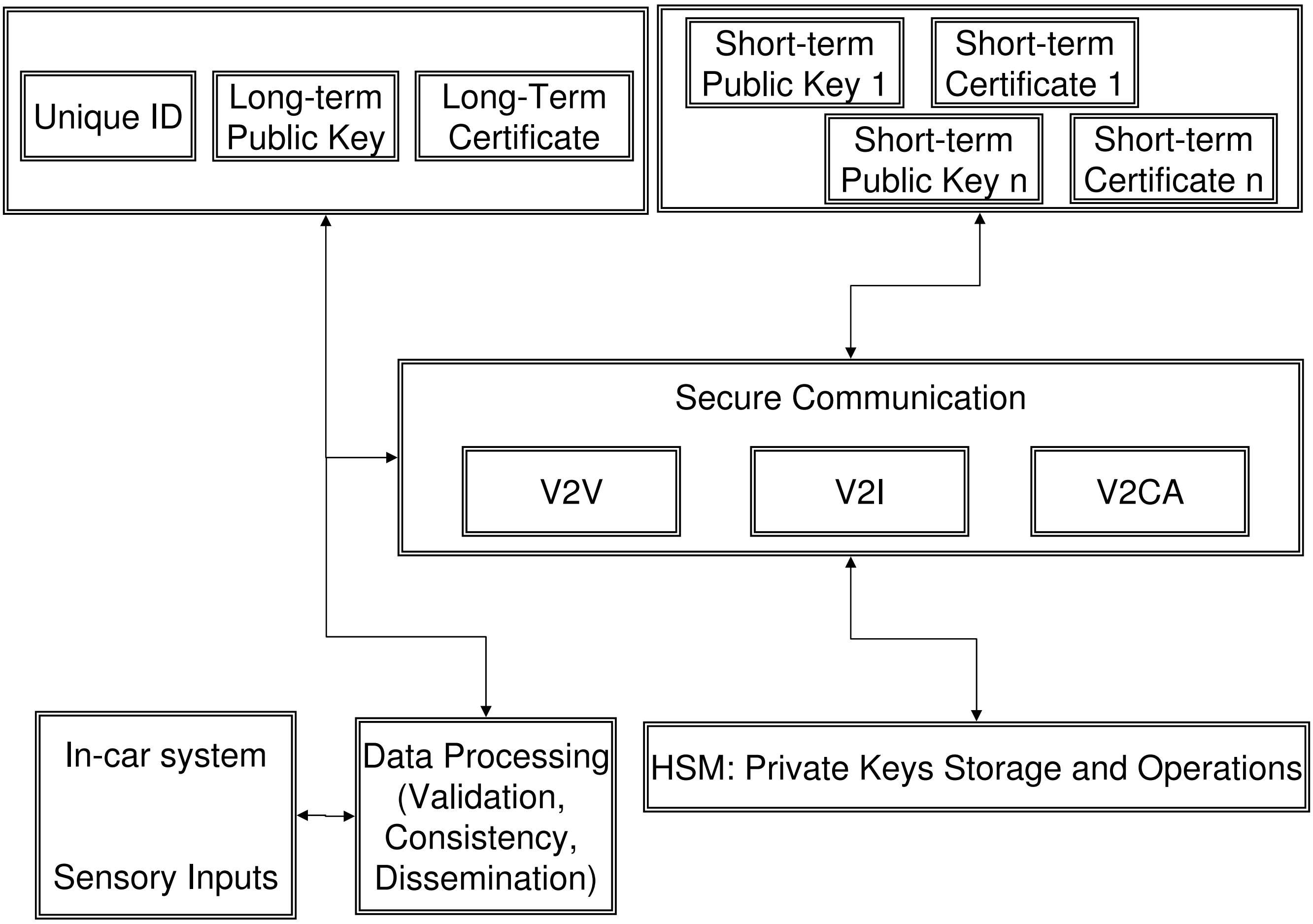

Fig. 2. Conceptual Secure VC Architecture View: Node functionality.

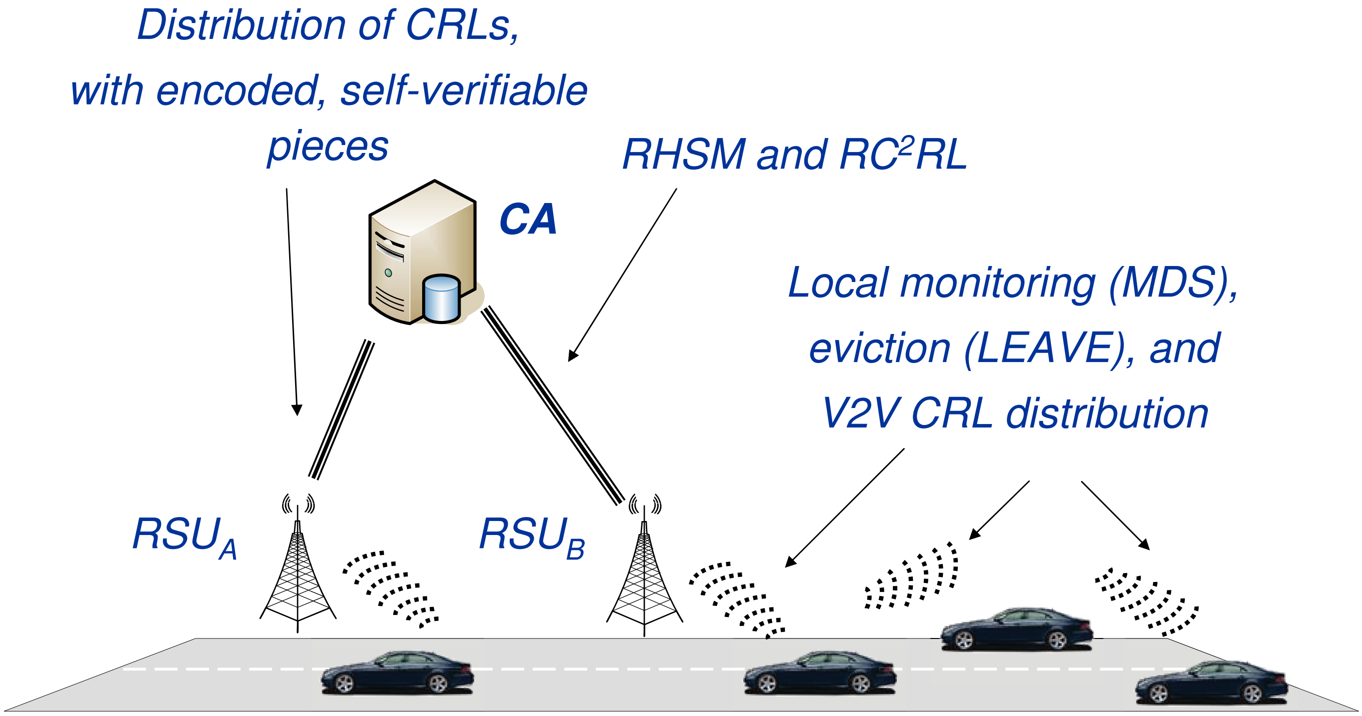

Fig. 3. Solutions of the Revocation Problem in VC Systems.

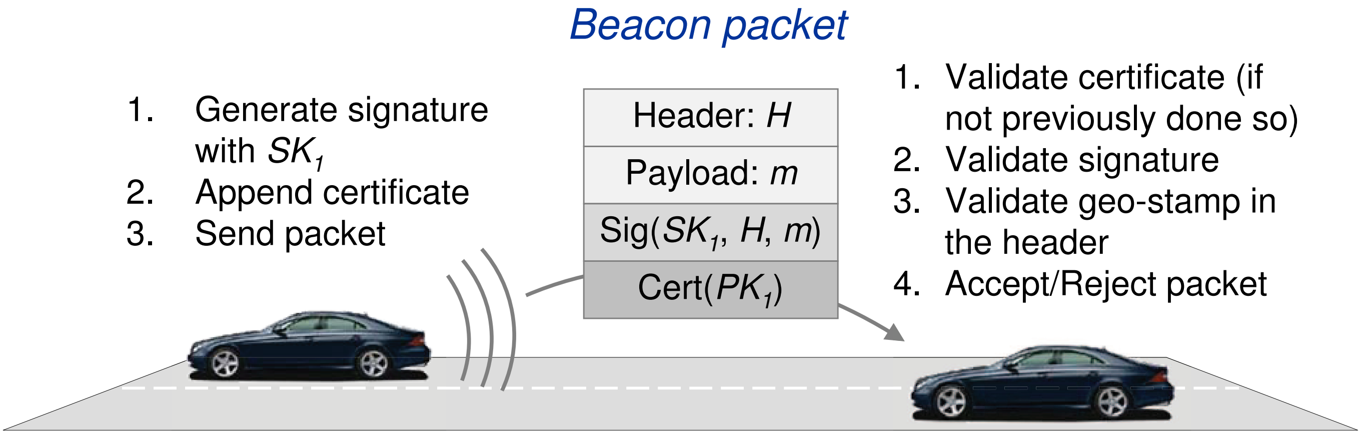

Fig. 4. Example of Secure Communication: Secure Beaconing

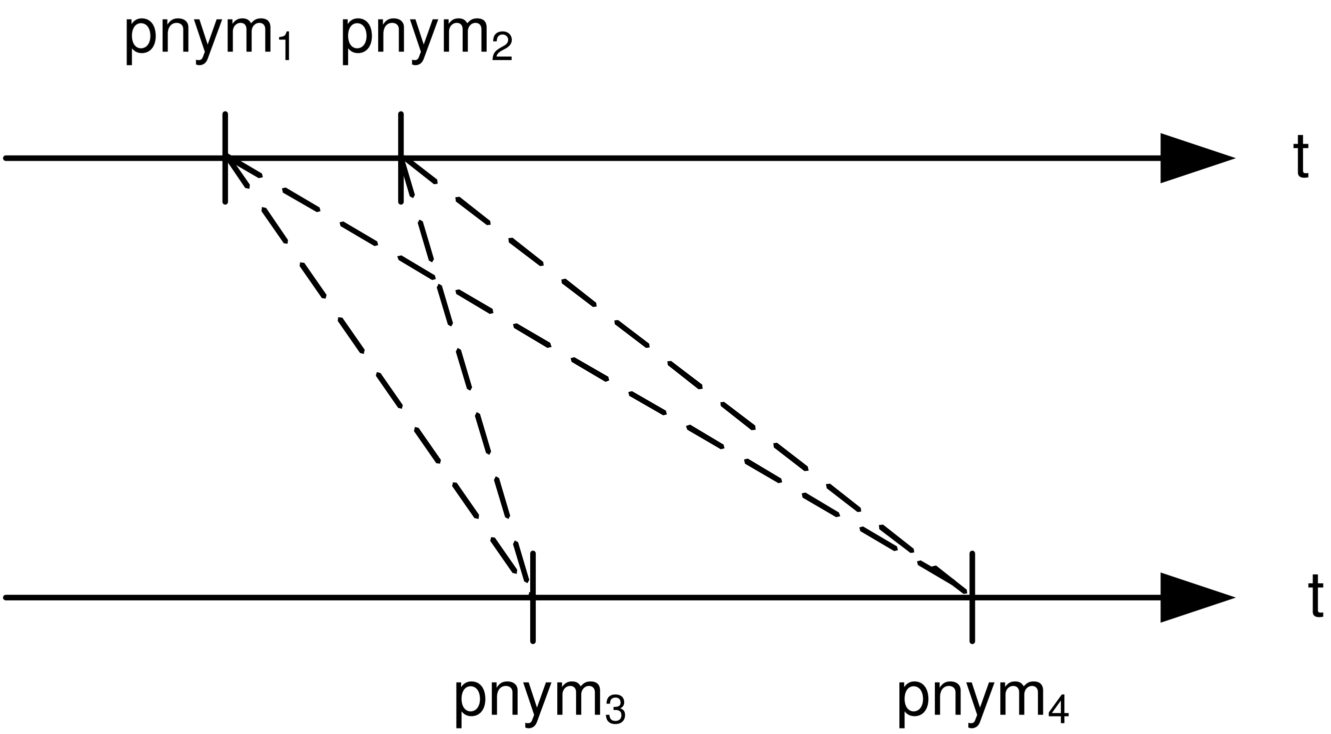

Fig. 5. Privacy Protection: Given vehicles entering and exiting a mix zone with different pseudonyms, the adversary must predict the most likely matching of event.