iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** XII **Month of publication:** December 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Wireless Vehicular Communications

M. Srividya[1], P.Sangeetha[2], MRS. N.Vijayarani[3]
[1+2]M.phil Full Time Research Scholar Department of Computer Science
[3]Assistant Professor Department of Computer Science & Applications
Vivekananda College of Arts and Science for Women (Autonomous), Namakkal, Tamilnadu, India

Abstract--- The deployment of vehicular communication (VC) systems is strongly dependent on their security and privacy features. We propose security architecture for VC. Developments took 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. The primary objectives of the architecture include the management of identities and cryptographic keys, 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 (http://www.sevecom.org), 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. A transversal project providing security and privacy enhancing mechanisms compatible with the VC technologies currently under development by all EU funded projects.
Index Terms—Vehicular networks, position verification, privacy, vehicular communication, sevecom

I. INTRODUCTION

After the deployment of various vehicular technologies, such as toll collection or active road-signs, vehicular communication (VC) systems are emerging. 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 toinfrastructure (V2I) communication can enable a range of applications to enhance transportation safety and efficiency, as well as infotainment. For example, they can send warnings on environmental hazards (e.g., ice on the pavement), traffic and road conditions (e.g., emergency braking, congestion, or construction sites), and local (e.g., tourist) information.

VC offer a rich set of tools to drivers and administrators of transportation systems but, at the same time, they make possible a formidable set of abuses and attacks. Consider, for example, nodes that 'contaminate' large portions of the vehicular network with false information, or the deployment of nodes that collect VC messages, track the location and transactions of vehicles and infer sensitive information about their drivers. Worse even, vehicles and their processing and sensing equipment can be physically compromised, while any wireless-enabled device could pose a threat to the VC system. These simple examples of exploits indicate that under all circumstances VC systems must be secured. Otherwise antisocial and criminal behavior could be made easier, actually jeopardizing the benefits of the VC system deployment. A comprehensive set of security mechanisms is thus critical and facilities and protocols that mitigate attacks are necessary.

A prominent example of those efforts is our three-year European-funded Secure Vehicular Communications (SeVeCom) Project (http://www.sevecom.org), which is approaching 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.

The baseline architecture relies on well-established and understood cryptographic primitives, which are already broadly implemented and scrutinized and thus deserve to be sufficiently trusted. At the same time, our architecture allows deployed systems to be tuned or augmented, in order to meet more stringent future requirements. We describe next the objectives and then the basic elements of our architecture and also implement the heat skin equipment. We conclude with a short discussion that ushers 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, comply with the implemented protocols, or they may deviate from the protocol definition, that is, be faulty or adversarial. Adversarial behavior can vary widely, according to the implemented protocols and the capabilities of the adversary. Its incentive may be own benefit or malice. We do not consider here 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, which can cause a much larger set of faults. We do not dwell on individual VC protocols for which to describe attacks. 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 will 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, that is, synthesize in a manner non-compliant to the protocols and system operation, and inject messages. Since 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, that is, interfere deliberately and prevent other devices within its range to communicate. 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.

Within this area, they can cause denial of service and do it 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 will encounter few or only a single adversary.

Due to the nature of VC systems, with vehicles equipped with a number of sensors, 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, is by controlling 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. AUTHORITIES

Drawing from the analogy with existing administrative processes and automotive authorities (e.g., city or state transit authorities), a large number of certification authorities (CAs) will exist. Each of them is responsible for the identity management of all vehicles registered in its region (national territory, district, county, etc.). Fig.1 illustrates a part of an instantiation of the CAs: an hierarchical structure within each CA and cross-certification among CAs. This way, the deployment of secure vehicular communications could still be handled locally to a great extent. At the same time, vehicles registered with different CAs can communicate securely

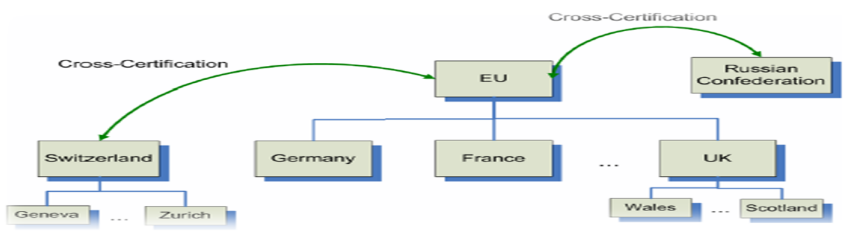


Fig. 1. Example of Hierarchical Organization and Relations of Certification Authorities.

as soon as they validate the certificate of one *CAA* on the public key of *CAB*. Various procedures for easily obtaining these cross-certificates can be implemented.

Nodes of the vehicular network are registered with exactly one CA. Each node, vehicle or RSU, has a unique identity *V* and a pair of private and public cryptographic keys, *kv* and *Kv* , respectively, and is equipped with a certificate *CertCA{V, Kv, Av, T}*, where *AV* is a list of node attributes and *T* the certificate lifetime. The *CA* issues such certificates for all nodes upon registration, and upon expiration of a previously held certificate.

We emphasize that the *CA* manages long-term identities, credentials, and cryptographic keys for vehicles. In contrast to short-lived keys and credentials as those discussed in Sec. IV. This issue is discussed in Sec.VI. The interaction of nodes with the *CA* does not need to be continuous, while the roadside infrastructure or other infrastructure-based networks (e.g., cellular) could act as a gateway to the vehicular part of the network or offer an alternative method of connectivity.

IV. SECURITY REQUIREMENTS

The problem at hand is to secure the operation of VC systems, that is, design protocols that mitigate attacks and thwart deviations from the implemented protocols to the greatest possible extent. Different protocols have their own specifications, that is, sought properties. Rather than providing an exhaustive enumeration of requirements per protocol and application, we identify first 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:

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 likeness of the sender. In other words, ascertain that a received unmodified message was generated within an interval [t_ ; t], with t the current time at the receiver and _ > 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.

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.

V. SECURE VC SYSTEM OVERVIEW

Our architecture addresses the following fundamental issues: (i) identity, credential and key management, and (ii) secure communication. We focus primarily on securing the operation of the wireless part of the VC system, and enhancing the privacy of its users, seeking to satisfy the requirements we outlined earlier in this article. 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 (CAA and CAB), 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 provides certificates for other CAs (cross-certification) or provides foreigner certificates to vehicles that are register with another CA when they cross the geographical boundaries of their region [10].

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. 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.

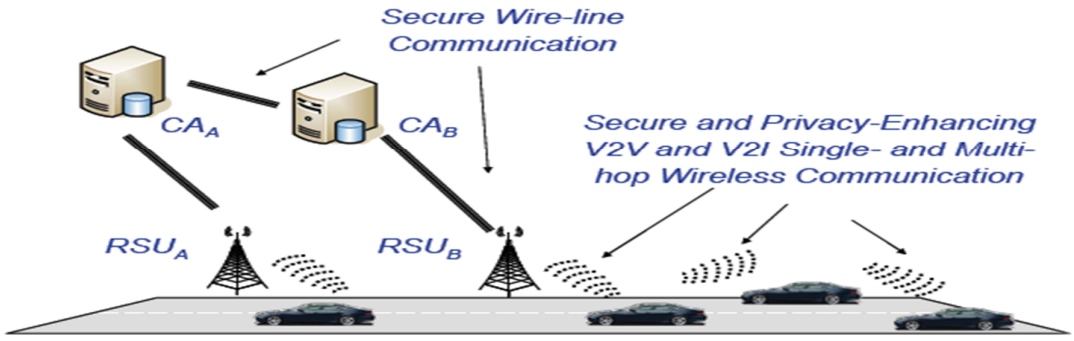


Fig.2. Abstract View of the Secure Vehicular Communication System.

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, which 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 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) also possible. The conceptual view of VC nodes is illustrated in 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 tool to secure communications, 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.

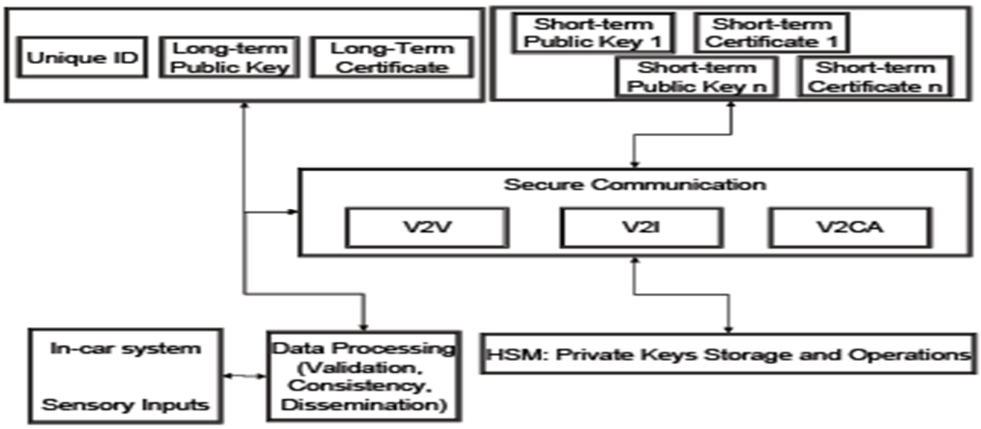


Fig. 3. Conceptual Secure VC Architecture View: Node functionality.

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 beacon messages for safety, 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.

VI. CREDENTIAL MANAGEMENT AND CRYPTOGRAPHIC SUPPORT

The management of credentials, both short and long-term, is undertaken by the CAs, which 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 IDX, 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 roadside units. Each identity is associated with a cryptographic key pair (SKX;PKX), 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 is naturally 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 f(SK1v ; PK1*v*); :::; (SKiv ; PKiv)g and sends the public keys to a corresponding CA via a secured communication channel. V utilizes its long-term identity IDV to authenticate itself to the CA. The CA signs each of the public keys, PKiv, 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 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 (SKj+1; PKj+1) is done, 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.

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.

By using the same pseudonym only for a short period of time and switching to a new one, vehicle activities can be only linked over the period of using the same pseudonym. 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.. An observer could then be successful with high probability in guessing 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. 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.

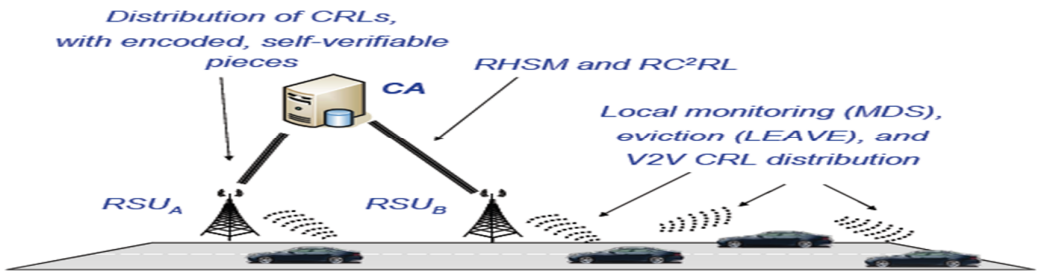


Fig. 4. Solutions of the Revocation Problem in VC Systems.

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 update 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, K1 and K2, 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, says K1, can be revoked, as discussed in the next paragraph. The revocation command must be signed with the private key corresponding to K1 itself. Once K1 is revoked, a new key K'1 can be loaded into the HSM by a command signed with the private key corresponding to K2. In addition, when K1 is revoked, the HSM does not accept commands aimed at revoking K2. 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, which 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.

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 mechanisms to achieve this are 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, 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, 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 (SKX). 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 RC2RL (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 RC2RL 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 omni-present 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

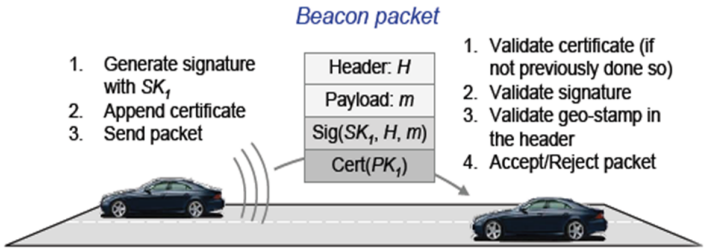


Fig. 5. Example of Secure Communication: Secure Beaconing

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.

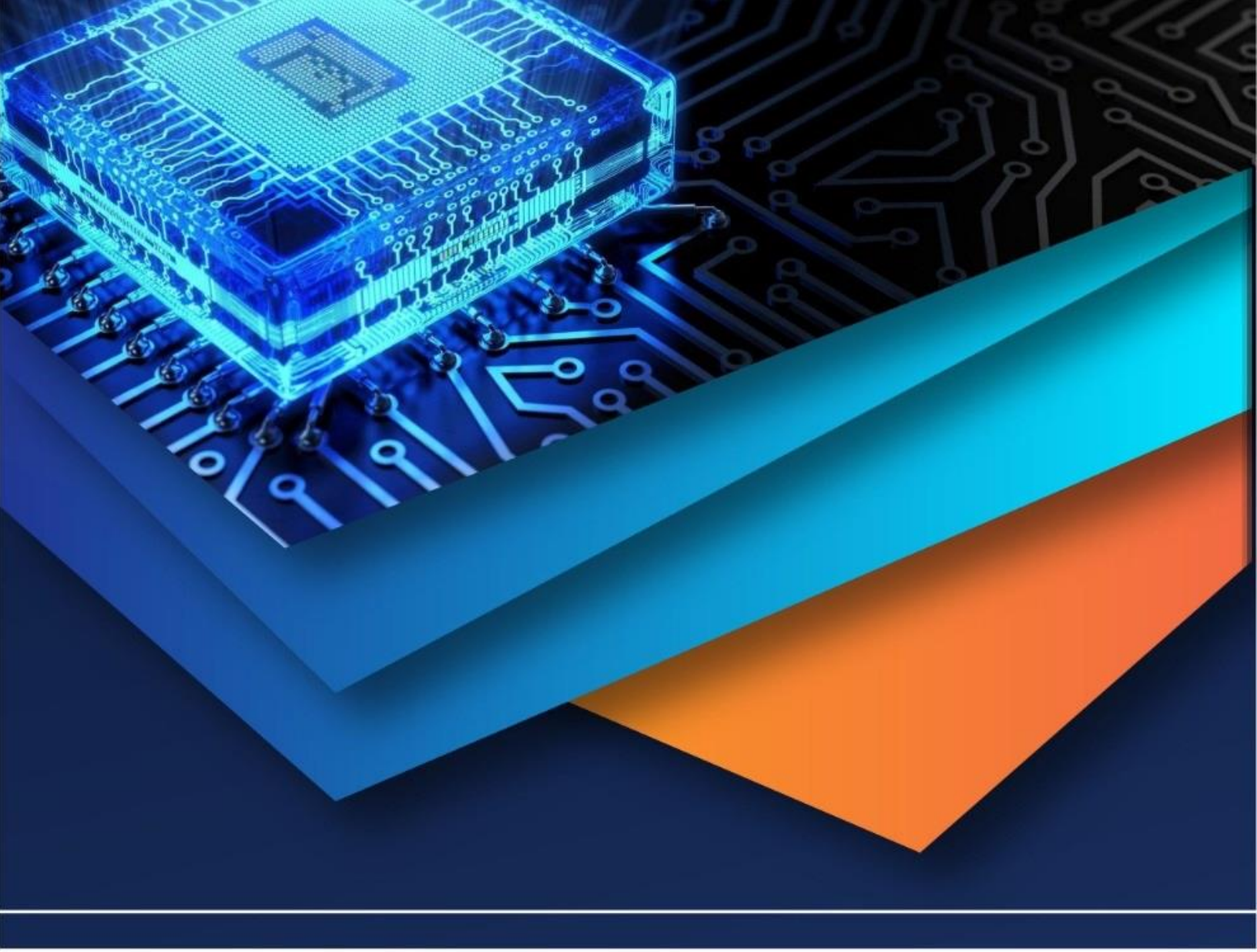
VII. CONCLUSIONS

We have developed a security architecture for VC systems, aiming at a solution that is both comprehensive and practical. We have studied the problem at hand systematically, identifying threats and models of adversarial behavior as well as security and privacy requirements that are relevant to the VC context. We 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 remaining research challenges towards further development and deployment of our architecture.

REFERENCES

[1] M. Gerlach, A. Festag, T. Leinmller, G. Goldacker, and C. Harsch, "Security Architecture for Vehicular Communication," 5th International Workshop on Intelligent Transportation (WIT), March 2007
[2] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in Proceedings of the 2004 Workshop on Vehicular Ad hoc Networks (VANET), 2004
[3] C. Harsch, A. Festag, and P. Papadimitratos, "Secure Position-Based Routing for VANETs," in Proceedings of the IEEE 66th Vehicular Technology Conference VTC2007-Fall, Baltimore, Oct. 2007 (to appear)
[4] R. Hauser, T. Przygienda and G. Tsudik, "Reducing The Cost Of Security In Link-State Routing," In Proceedings of the Symposium on Network and Distributed System Security, 1997
[5] http://sumo.sourceforge.net/
[6] http://www.isi.edu/nsnam/ns/
[7] http://wiki.ep_.ch/trans
[8] F. Kargl, S.Schlott and M. Weber, "Identi_cation in Ad hoc Networks," Hawaiian International Conference on System Sciences (HICSS 39), January 2006
[9] F. Kargl, Z. Ma, E. Schoch, "Security Engineering for VANETs," 4th Workshop on Embedded Security in Cars (escar 2006), November 2006
[10] K. Laberteaux and Y.-C. Hu, "Strong VANET Security on a Budget," In Workshop on Embedded Security in Cars (ESCAR), 2006
[11] R. Mangharam, and D. Weller, D. Stancil, R. Rajkumar and J. Parikh, "GrooveSim: a topography-accurate simulator for geographic routing in vehicular networks," In Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks (VANET'05), 2005
[12] A. Saha and D. Johnson, "Modeling mobility for vehicular ad-hoc networks," In Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks (VANET'04), Poster session, 2004
[13] C. Tchepnda, H. Moustafa, H. Labiod and G. Bourdon "Securing Vehicular Communications: An Architectural Solution Providing a Trust Infrastructure, Authentication, Access Control and Secure Data Transfer," In Proceedings of the 1st IEEE Workshop on Automotive Networking and Applications (AutoNet'06), 2006



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY