Secure Vehicular Communication Systems: Implementation, Performance, and Research Challenges

Frank Kargl, Ulm University
Panagiotis Papadimitratos, EPFL
Levente Buttyan, Budapest University of Technology and Economics
Michael Müter, Daimler AG
Elmar Schoch and Björn Wiedersheim, Ulm University
Ta-Vinh Thong, Budapest University of Technology and Economics
Giorgio Calandriello, Politecnico di Torino
Albert Held, Daimler AG
Antonio Kung, Trialog
Jean-Pierre Hubaux, EPFL

ABSTRACT

Vehicular communication systems are on the verge of practical deployment. Nonetheless, their security and privacy protection is one of the problems that have been addressed only recently. In order to show the feasibility of secure VC, certain implementations are required. In [1] we discuss the design of a VC security system that has emerged as a result of the European SeVeCom project. In this second article we discuss various issues related to the implementation and deployment aspects of secure VC systems. Moreover, we provide an outlook on open security research issues that will arise as VC systems develop from today's simple prototypes to full-fledged systems.

INTRODUCTION

Vehicular communication (VC) systems will enable many exciting applications that will make driving safer, more efficient and more comfortable. But this necessitates the introduction of security and privacy enhancing mechanisms, as discussed in [1]. In this article we focus on the practical aspects associated with the implementation and deployment of such a secure VC system. We also provide an outlook to future research challenges.

First, we explain why the deployment of a security system for a vehicular environment is different compared to other common information technology systems. Then we present the SeVeCom baseline architecture, and highlight various implementation- and deployment-specific aspects such as flexible integration in existing communication stacks, use of a hardware security module, and secure connections of VC onboard units to in-vehicle bus systems. Furthermore, we analyze the performance and communication overhead of the suggested security mechanisms and propose optimizations for efficient secure communication.

Finally, we present selected topics we consider relevant for future research on VC system security. One aspect is the use of complex forms of data dissemination, such as aggregation schemes, which require different security approaches than those used for broadcast and unicast communications. Another aspect is integrating VC systems with other networks or connecting them with mobile commodity devices, which raise additional security problems. Other future research aspects include secure localization and discovering whether existing VC privacy solutions are indeed sufficient.

VEHICULAR COMMUNICATION SYSTEMS

There are significant differences between devices such as mobile phones or desktop computers connected to the Internet and devices in a VC system. Differences in development, production, and operation determine VC-specific constraints and conditions:

0163-6804/08/$25.00 © 2008 IEEE

- Vehicles have a long life span, lasting several years in most cases. This makes it hard to change onboard systems in order to mitigate new risks to the vehicle safety.
- Owners have constant physical access to and full control over their vehicles. In spite of the involved safety risks, many users might try to modify or "enhance" their vehicles. From a manufacturer's point of view, the risk of hardware tampering cannot be neglected.
- No technical expertise in vehicle electronics or VC security aspects is expected from a user who runs a vehicle. Hence, the vehicular security measures have to operate autonomously with no need for intervention or feedback from the user.
- Robustness requirements and time constraints are demanding. Functions necessary, for example, for driving or alerts received via the VC system must be processed in real time: delays or errors could lead to vehicle malfunctions, driving errors, and consequently to physical damages and injuries.
- Liability and conformance require precise formulation of legal issues. As regulations and requirements differ from country to country, it is even more difficult to address these challenges.

These observations have consequences on the implementation of a VC security system. Due to the long life cycle of vehicles, it cannot be ensured that all threats are thwarted at the time of development. Therefore, the VC security mechanisms should be flexible, adaptable, and extensible, to allow adjustments to changing security requirements. To address this need, we propose a component-based security architecture for VC systems, which allows to us add, replace, and reconfigure components (e.g., substitute cryptographic algorithms) throughout the life cycle of the vehicle.

The large number and variety of vehicles have to be taken into account. Even for a single car type, different production and equipment lines lead to many distinct versions and variants. Nonetheless, it should be possible to integrate a security system into all those platforms. In addition, the communication stack and security measures might be designed by different teams or vendors; a situation that clearly requires well defined but still flexible interfaces. These reasons led to the development of the so-called *hooking architecture*, which introduces special hooks at the interface between every layer of the VC system. The hooking architecture introduces an event-callback mechanism into the communication stack, which allows adding security measures without the need to change the entire communication system. The security system in a vehicle has to fulfill real-time or near-real-time requirements. For the underlying cryptographic primitives, this implies optimized cryptographic hardware in order to guarantee near-real-time performance. The potential trade-off between security and performance has to be well balanced.

To enable VC systems to withstand future, yet unknown attacks, besides the traditional prevention-oriented approach, functionalities to detect attacks (such as intrusion detection capabilities) and to recover after attacks are needed. In the long run, the goal is to enhance the resilience of the system.

To enable VC systems to withstand future, yet unknown attacks, besides the traditional prevention-oriented approach, functionalities to detect attacks (such as intrusion detection capabilities) and to recover after attacks are needed. In the long run, the goal is to enhance the resilience of the system.

SeVeCom Implementation

The SeVeCom project defines a baseline security architecture for VC systems [2]. Based on a set of design principles, SeVeCom defines an architecture that comprises different modules, each addressing certain security and privacy aspects. Modules contain components that implement one part of system functionality. The baseline specification provides one instantiation of the baseline architecture, building on well-established mechanisms and cryptographic primitives, thus being easy to implement and to deploy in upcoming VC systems.

Baseline Architecture: Deployment View

The SeVeCom baseline architecture addresses different aspects, such as secure communication protocols, privacy protection, and in-vehicle security. As the design and development of VC protocols, system architectures, and security mechanisms are an ongoing process, only a few parts of the overall system are yet finished or standardized. As a result, a VC security system cannot be based on a fixed platform but instead has to be flexible, with the possibility to adapt to future VC applications or new VC technologies.

To achieve the required flexibility, the SeVeCom baseline architecture consists of modules that are responsible for a certain system aspect, such as identity management. The modules, in turn, are composed of multiple components, each handling a specific task. For instance, the secure communication module is responsible for implementing protocols for secure communication and consists of several components, each of them implementing a single protocol. Components are instantiated only when their use is required by certain applications, and they use well defined interfaces to communicate with other components. Thus, they can be exchanged by more recent versions, without other modules being affected.

As shown in Fig. 1, the *security manager* is the central part of the SeVeCom system architecture. It instantiates and configures the components of all other security modules and establishes the connection to the cryptographic support module. To cope with different situations, the security manager maintains different policy sets. Policies can enable or disable some of the components or adjust their configuration, for example, to enhance or relax the parameters for a pseudonym change under certain circumstances.

Communication Stack Integration

To be independent of the actual communication stack, the integration of the SeVeCom security system into the protocol stack is based on a *hooking concept*, inspired by similar architectures such as the Linux Netfilter kernel subsystem. Interlayer proxies (ILPs) are inserted at several points in the communication stack. Every ILP

Whenever the SeVeCom system is ported to a new platform, besides adapting to different packet formats, only the ILPs and the convergence layer have to be modified, while all other components remain unaffected both in terms of security and communication.

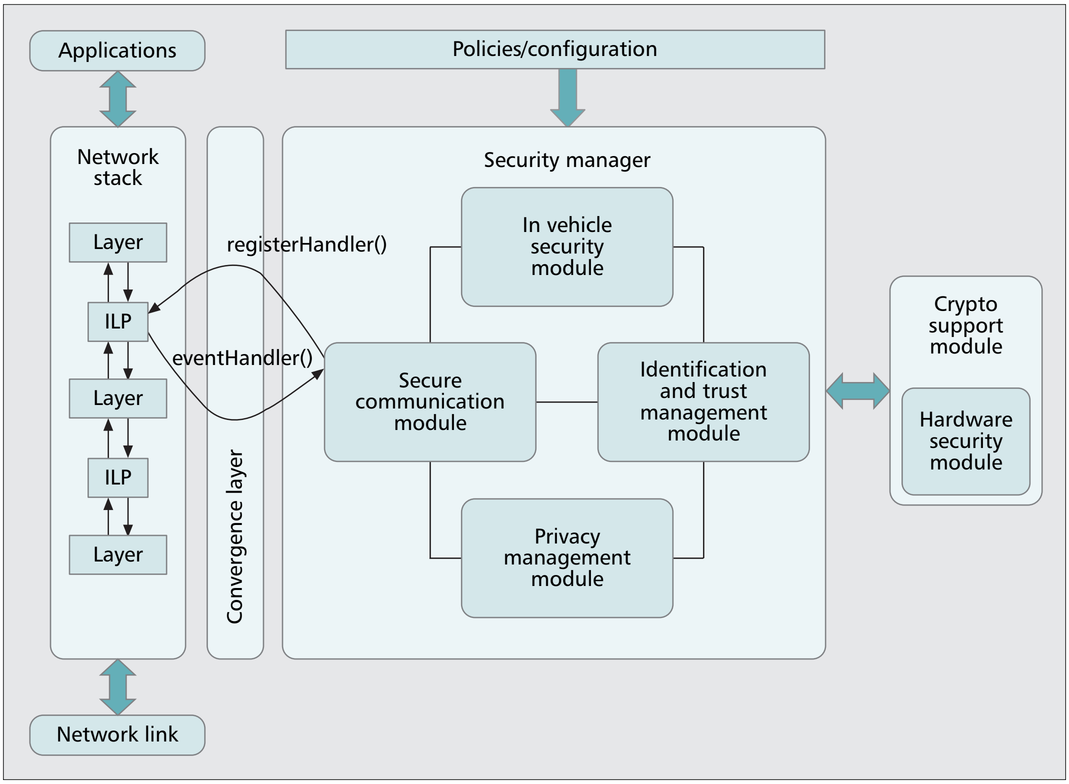


Figure 1. *Baseline architecture: deployment view.*

maintains a list of callback handlers that are to be notified of certain events.

During initialization, the SeVeCom components can register at an ILP, subscribing for certain message types and direction (up or down the stack). Therefore, they have to implement an event listener interface and use the *registerHandler()* method to connect to an ILP. Some components may have to register at multiple ILPs, subscribing for different kinds of packets. When a message arrives at an ILP, an event callback is triggered for all components that have registered for this message type and their *eventHandler()* method is called. The callback includes a reference to the received message, and the component is then able to inspect or modify it. With the return value, the component indicates if the message was modified, if it should be reinserted into the stack, or if it should be simply dropped by the ILP. The secure beaconing component, for example, connects to the ILP above the medium access control (MAC) layer and checks the signatures of all incoming beacon messages. Beacons with invalid signatures are either discarded or tagged. Using this hooking architecture, it is possible to transparently integrate security functionality into an existing network stack with minimal modifications. Whereas events are triggered by the communication stack, the security system can also access the stack by means of command calls using a well defined application programming interface (API) offered by stack layers. Command calls could, say, instruct the MAC layer to set its MAC address to that of a new pseudonym.

The hooking concept makes certain assumptions about the network stack. It assumes a layered architecture, where the ILPs can be inserted in between, and the stack has to implement a certain command API (e.g., for change of MAC addresses). To be able to port the SeVeCom architecture to many different communication platforms, we also provide an additional convergence layer. This defines an abstraction interface that proxies call between the communication system and the security components. Whenever the SeVeCom system is ported to a new platform, besides adapting to different packet formats, only the ILPs and the convergence layer have to be modified, while all other components remain unaffected in terms of both security and communication.

Hardware Security Module

As explained in [1], the purpose of the hardware security module (HSM) is to provide a physically protected environment for the storage of private keys and the execution of cryptographic operations using them. Clearly, the full implementation of an HSM is beyond the scope of the SeVeCom project, but we can summarize the main requirements that such an implementation should meet in order to be applicable to securing VC systems.

First of all, the HSM must be tamper-resistant to some extent. High-end tamper-resistant modules (e.g., the IBM 4758 Cryptographic Coprocessor) are too expensive to be added to every vehicle. At the same time, we observe that low-end tamper-resistant devices (e.g., smart cards) do not provide all the functionality we need. In particular, commercially available low-end devices do not have built-in batteries and consequently cannot provide a trusted internal clock. As pointed out in [3], without a trusted source of time, such devices are not able to produce timestamps that can be trusted by other participants in the system. Therefore, we need an HSM implementation somewhere between high-end and low-end devices. A potential approach is to implement the HSM as an application-specific integrated circuit (ASIC) with some special coating that provides a certain level of tamper resistance. Such a customized device can provide all the necessary functionality by design, and it can be produced in large quantities at sufficiently low costs.

Second, the HSM must have an API through which it can provide services to the other modules of the security architecture that run on the onboard unit (OBU). This API should support the digital signature and timestamping service, the decryption service, as well as the key and device management services described in [1]. We specify such an API in the SeVeCom project; however, lacking the appropriate HSM hardware, we only implement it in the form of a software library running on a general-purpose computer. Nevertheless, besides being useful for demonstration purposes, our implementation can also serve as a reference for future implementations on real HSM devices. In our implementation we use ECDSA for digital signature generation, and ECIES with HMAC-SHA1 and AES-CBC for encryption, and we fully implement the key management services of the HSM described in [1].

Finally, we note that some examples published in [4] show that physically secure modules can successfully be attacked through their weakly designed APIs. For this reason, we use formal verification techniques to verify the SeVeCom HSM API. Our method is based on the applied pi-calculus and an automated verification tool called *ProVerif*. We prove that a key generated by an adversary cannot be implanted as a new root key in the HSM through the API. Additionally, short-term and long-term private keys are proven not to be revealed after series of function calls.

IN-VEHICLE SECURITY

In order to achieve their full potential, VC systems need access to the in-car network and sensors that observe the current status of the vehicle and the environment. This enables a VC system to process signals such as emergency braking, airbag activation, and slippery road detection, thus greatly contributing to the avoidance of accidents and improvement of road safety.

Onboard system signals are transferred inside the car through different networks and domains. Usually, the network architecture and in-car gateways restrict the signals to the defined network segments and prevent information from leaving its dedicated domains. This clear architecture and strict separation is one measure that ensures the entire vehicle always, especially its vital functions (brakes, engine or airbag control), operates reliably and *cannot* be attacked from the outside. If this were to be changed into a more open architecture, for example, by allowing for reading out sensor information from in-vehicle networks or displaying and reacting to warning messages from external sources, it would be absolutely necessary to ensure that in-vehicle systems are protected from any external malicious influence.

The *in-vehicle security module* protects the interface between in-car networks and the wireless communication system. It controls external access to in-car networks, onboard control units, and vehicle sensor data, but it also ensures that data and services required by other V2V and V2I applications are provided correctly. Within the in-vehicle security module, two main components are provided:

- A *firewall* that controls the data flow from external applications to the vehicle and backwards
- An intrusion detection system (IDS) that constantly monitors the status of in-car systems and provides real-time detection of attacks

The firewall realizes a packet- or application-based firewall approach. Its rule-based table states which application is allowed to access each kind of data or service. The IDS can dynamically add rules to the firewall table in order to deny access for a specific application or disable a service.

The IDS is based on an anomaly detection approach, which implies that normal onboard system behavior is clearly defined and specified. If an event results in an onboard system state that is not part of the standard specification, a potentially dangerous situation is detected. Depending on the source and type of event, appropriate reactions are taken to get the system back to a secure and safe state.

The firewall realizes a packet or application based firewall approach. Its rule-based table states which application is allowed to access each kind of data or service. The IDS can dynamically add rules to the firewall table, in order to deny access for a specific application or disable a service.

PERFORMANCE ISSUES

One very important aspect toward deployment is performance. Given cost constraints in today's car manufacturing, one cannot equip vehicles with state-of-the-art desktop processors. Instead, inexpensive and energy-saving embedded processors are used. At the same time, cryptographic operations to secure VC [1] create significant overhead in terms of both processing and communication bandwidth.

This is especially true because beaconing is a fundamental VC protocol: vehicles frequently send information (e.g., position and environment conditions), typically 1 beacon/100 ms. At these rates, the security overhead would always be significant. 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 ms, beacons from *all* neighboring vehicles in range, which, not to forget, may also change their identity (pseudonym) in the meantime.

Although RSA and DSA signatures have long been industry standards, these mechanisms do not meet the overhead requirements in terms of either processing or bandwidth. Especially in combination with large X.509v3 certificates, they are unsuitable for high-speed and low-overhead

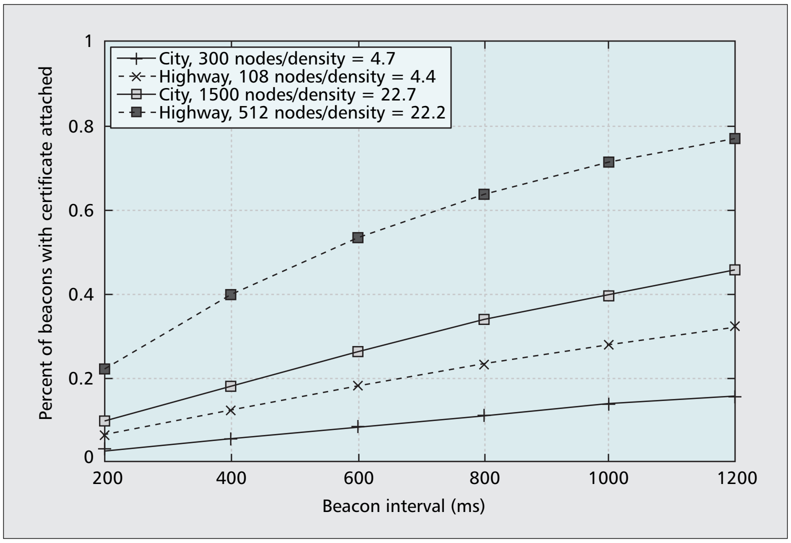


Figure 2. *Performance results: beacon certificate omissions ([7]).*

VC systems. In contrast, for the same security levels Elliptic Curve Cryptography (ECC), including ECC signatures, keys, and certificates, is significantly smaller than its RSA and DSA counterparts. This is the reason SeVeCom, as well as the IEEE 1609.2 trial standard, chose to utilize EC-DSA signatures. In addition, SeVeCom utilizes compact certificates.

To reduce overhead, [5, 6] propose to *not* attach certificates to all messages, but rather for one every α successive beacons; they also propose certificate caching to reduce verification processing overhead. Additional optimizations are proposed in [7]: omitting signatures or signature verifications in certain situations, and avoiding attaching certificates based on the context, that is, *unless* a change in the vehicle neighborhood takes place.

Such overhead can affect VC applications in multiple ways. An investigation on safety applications is provided by [5, 6]. 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: each receiver *V* must in principle verify a signature for each received packet, whereas signature generation is not as critical (in general, *V* signs one and verifies *N* messages per time slot). Simulations in [6] show that the CPU of a vehicle is heavily stressed in situations with dense topologies (e.g., in congested multilane highways) even if vehicle direction is used to avoid processing messages from vehicles in the opposite flow.

These findings assume hardware for OBUs that are used for current VC prototypes; in upcoming field trials OBUs are expected to have less powerful hardware, such as a Power PC CPU at 400 MHz [8]. Initial products presumably will be equipped similarly. The actual crypto performance of this hardware depends very much on the implementation (e.g., if precalculated tables are used), but assuming efficient software libraries and ECDSA-224, we estimate that this hardware will not be able to process more than a few dozen verifications per second. Dedicated ASICs are expected to be able to handle the required cryptographic load at moderate costs [9].

By integrating optimization mechanisms to pseudonymous authentication, such as those of [5–7], we introduce additional verification delays: safety warnings can be trusted only if the corresponding short-term certified public key (pseudonym) was previously verified. It turns out that the overall effect of security on this front can be kept low, with the number of crashes experienced in a platoon of vehicles close to that achieved without any security mechanism [5, 6]. An additional mechanism of repeatedly attaching a certificate to β successive beacons when a pseudonym change takes place can increase reliability. We note, however, that the performance of safety applications is heavily influenced by other parameters, like the placement of vehicles, the beaconing rate, and the penetration rate of vehicular communication.

Based on the optimizations discussed above, Fig. 2 shows the fraction of beacon messages with attached certificates in various scenarios and several beaconing intervals. Certificates are attached to beacons only if new neighbors are discovered. The result is that certificates can be omitted in more than 90 percent of all sent beacons, for small beacon intervals and medium node density. Figure 3 shows the impact of security on safety for an emergency braking notification application in a challenging, dense, fast-moving network: with 1/α the fraction of sent certificates and β the number of repetitions upon pseudonym change, the effectiveness of unsecured VC is practically the same as that of secure VC.

Research Challenges

We consider our VC security solution, described here and in [1], mature and practically deployable. Nonetheless, there are open issues that cannot be handled by existing security strategies alone, thus calling for new approaches. We highlight those, present initial results, and raise questions toward future research.

Alternative Communication Forms

Ongoing research has mostly considered VC protocols relying on periodic beaconing, flooding, Geocast, and position-based routing. Up to now, these mechanisms have received the attention of work on VC security and privacy. Nonetheless, recently, additional means of information dissemination have been considered in the context of VC. For example, the literature highlights the need for more efficient flooding and Geocast strategies, and suggests the use of gossiping or context-adaptive message dissemination, as well as data aggregation in VC systems [10].

These new approaches will necessitate an adaptation of security and privacy strategies. Mechanisms such as context-adaptive message dissemination already provide an inherent degree of resistance against attacks [11]. In contrast to many routing protocols, where the protocol itself can become the target of an attacker, there is (almost) no signaling between nodes an attacker could exploit.

Another aspect that brings forth new security issues is the need for nodes that relay messages to also modify them. This has already been the case for position-based forwarding discussed in [1], but is even more so for context-adaptive message dissemination and data aggregation. In the latter case, individual information contributed by vehicles usually becomes unavailable during the dissemination process.

Misbehavior, including injection of erroneous data and denial of service (DoS), is still possible, even if strong cryptographic security is present. Mechanisms that perform consistency checks, using redundant information or onboard sensors, can be used to discard incorrect information from the network. Meanwhile, rate limits can confine the effects of DoS attacks. Initial exploration of such mechanisms has already delivered promising results [11].

Data-Centric Trust

We observe that the trustworthiness of messages sent by a node (vehicle or roadside unit, RSU) is primarily determined by the trustworthiness of the sender's credentials. Essentially, the VC system entities, certificate authorities (CAs), and node make statements on public keys, identities, attributes, data, and VC messages, respectively. Then, at any point in time, messages from any newly encountered car are trusted as long as their certificates are valid. Such trust relations, entity-centric and set a priori, are useful, but they lack the flexibility necessary for highly volatile and data-centric VC systems.

Given the majority of VC applications, it becomes clear that it is often more useful to assess the trustworthiness of data per se than to assess only the trustworthiness of the nodes that report them. The need for *data-centric trust establishment* is clearer if we consider that identities of nodes are largely irrelevant, even if no privacy enhancing mechanisms are employed. In contrast, it is the data (e.g., safety warnings, traffic information) and their freshness and location relevance that matter the most. From another point of view, trying to interact with possibly adversarial (faulty) data senders to determine their trustworthiness is hard: encounters are in general short-lived and have no prior association.

Considering more generally the issue of non-cryptographic protection mechanisms, it is possible to rely on its own or trusted measurements (e.g., as discussed in [1] for securing Geocast or in the previous section for context-aware data dissemination) to discard erroneous data. But data may come from relatively remote sources. More important, the receiving node will be unable to determine their trustworthiness alone. So cooperative management of data-centric trust is needed. Beyond what is presented in [12], further development of techniques to achieve data-centric trust establishment will be needed.

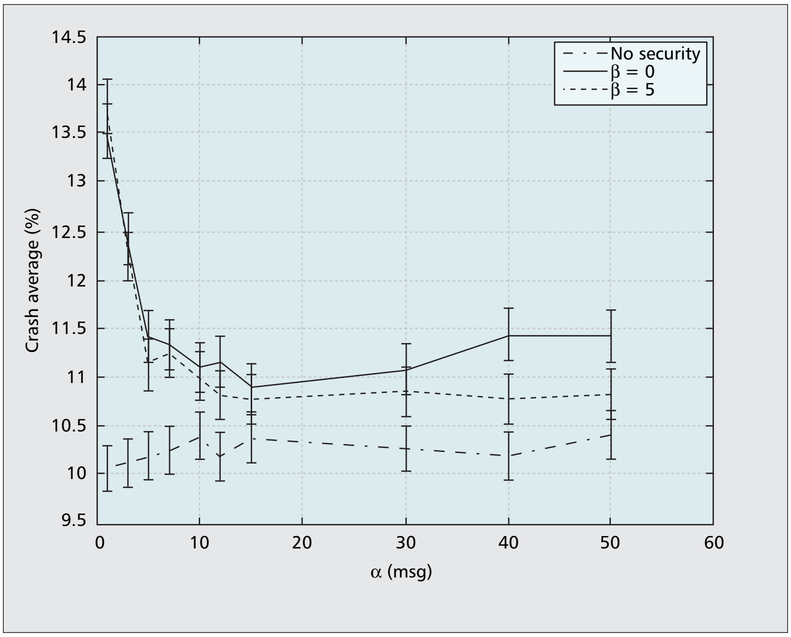


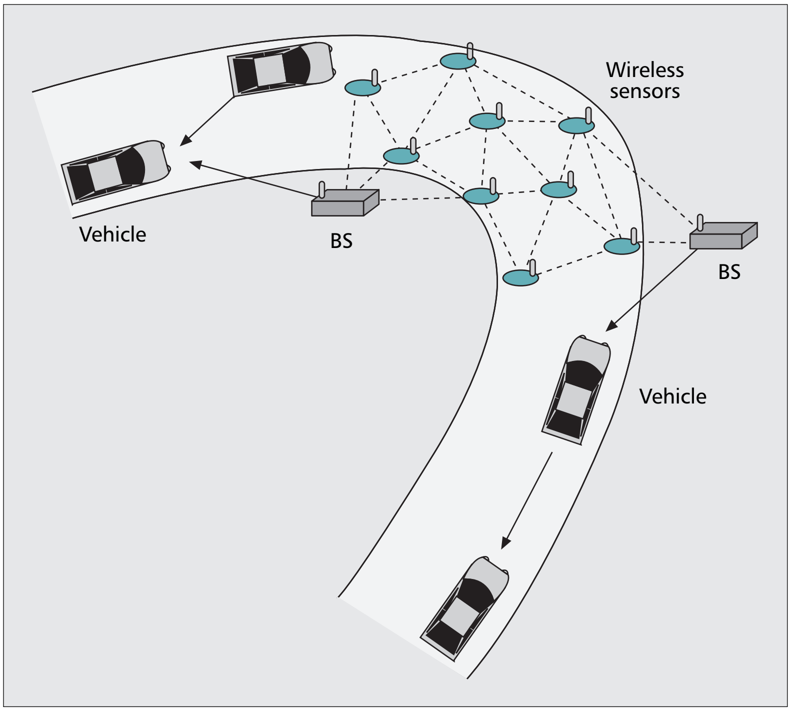
Figure 3. *Performance results: impact of security and privacy on safety ([6]).*

Secure Localization

Location information is critical for VC systems, especially for cooperative awareness, collision avoidance, and essentially all safety applications, as well as for position-based information dissemination. An internal adversary could falsely announce its own positions, and an input-controlling adversary [1] could affect the position announced by its victims, and this way disrupt or abuse VC operation. Whereas an internal adversary could be thwarted by data consistency checking and position verification, these methods cannot be effective against an input-controlling attacker that attacks the global navigation satellite systems (GNSS).

The objective of the adversary is to manipulate the location GNSS receivers compute, for example, for the global positioning system (GPS). To do so, the adversary can interfere with GNSS transmissions and inject forged navigation messages. A variant of such attacks, the replay attack, is possible even if GNSS were cryptographically protected. In fact, replay attacks can be fine-grained, so gradual manipulation of each victim location can remain small and thus hard to detect. But cumulatively, they can lead to substantial distances between the actual and perceived (provided by the GNSS) location of the victim nodes [13]. Equally interesting, such attacks are possible without any compromise, physical or otherwise, of the GNSS receiver or other onboard equipment or software.

This leads to an important realization: location information in the system cannot be considered trustworthy by default. One solution would be to leverage on mechanisms such as the secure neighbor discovery and position verification discussed in [1]. In addition, dedicated infrastructure could provide "landmarks," assisting the detection of false location information. Finally, mechanisms that detect adversarial GNSS transmissions could be devised and inte-



■ **Figure 4.** *Hybrid vehicle communication system.*

grated in the GNSS receivers or OBUs. If so, correct nodes falling prey to an input-controlling adversary would declare their own location information as faulty, and thus refrain from disseminating any false data in any messages they transmit. We believe that future efforts in these two main directions should be undertaken.

SECURE INTEGRATION OF COMMODITY DEVICES

Devices such as portable computers, mobile phones, iPods, or (portable) navigation systems are becoming widespread. Customers wish to use these devices inside of the vehicle, and connect them to the vehicle electronics where meaningful. Nowadays, mobile phones and iPods can already be connected to vehicles to some degree. In the future, complete and seamless integration is desired.

Portable navigation systems, for instance, could be improved by transferring data from the vehicle's rotation sensors of the wheels and the current velocity to improve navigation in tunnels. For the calculation of the route and arrival time, additional internal data (e.g., fuel status) could be taken into account. If the customer were allowed to connect her mobile computer to the vehicle network in an uncontrolled way, she could be given the chance to check the vehicle status in detail and change at will settings such as the engine configuration or the visual layout of the telematic system user interface.

Every interface and connection of non-VC devices to an in-vehicle system poses a threat and increases the risk that malicious code or adversaries gain access to the in-vehicle system. Wireless interfaces raise additional concerns, as illegitimate access could be easier and achieved from a distance. To prevent in-vehicle and thus VC system compromise, it is necessary to define specific policies that describe and devise security mechanisms that enforce parsimonious access of commodity devices to in-vehicle resources.

HYBRID VEHICLE COMMUNICATION SYSTEMS

VC systems could be integrated with other communication networks, such as cellular, WiFi, wireless sensor, and mesh networks. They could, for instance, take advantage of the ubiquitous coverage provided by cellular networks, especially in the initial deployment phase when their penetration rate is expected to be low. Beyond the obvious use of cellular data services for information download, including security related data, the cellular infrastructure could also be used for geocasting traffic and safety related information with less stringent delay requirements; systems in this direction have been investigated, for example, by the European Com2React project [14]. WiFi networks could be used for similar purposes — at higher data rates — in urban areas where WiFi coverage is substantial. Wireless sensor networks deployed along hazardous roads can collect and process local environmental information they share with vehicles passing by. Figure 4 shows an example of such a hybrid VC system which also incorporates sensor network nodes that deliver sensing data via a base station to nearby vehicles.

In terms of security, the integration of VC systems with other communication networks requires at least an integrated authentication infrastructure. In particular, vehicles need to authenticate the messages they receive from cellular base stations, WiFi access points, and roadside sensors before trusting and acting on them. Although integration with cellular networks seems to be straightforward by adopting the well established security mechanisms of those networks, the integration with WiFi and sensor networks is more challenging. One problem is roaming across WiFi access points, because different operators may support different authentication methods, and even if a common method is accepted (e.g., based on the TLS Handshake Protocol), a large-scale international PKI needs to be available. In case of wireless sensor networks (WSNs), the challenges bear similarities to those for VC systems, with additional problems stemming from sensor node resource constraints and lack of physical protection. Even if sensor nodes could authenticate themselves to vehicles, they could have been tampered with and compromised, and therefore their data may not be trusted. In fact, WiFi and WSN operators may not be trusted to the same extent as cellular network operator to not misuse (e.g., share) transactional data sensitive for users' privacy.

PRIVACY

Taking privacy concerns into account when designing VC systems is important for several reasons. First of all, privacy is a basic right, and we believe that new technologies should be designed in such a way as to make it possible to retain this right. In addition, the protection of privacy is made mandatory by laws in many developed countries.

For these reasons, we integrated a baseline privacy protection mechanism into our architecture based on using and changing pseudonyms. However, the proposed pseudonym mechanism has some limitations. It might still be possible to fully track vehicles between pseudonym changes. Increasing the frequency of changes can help, but also increases the incurred overhead. In addition, taking into account statistical models of the traffic in a given geographical area, tracking of vehicles is possible to some extent despite frequent pseudonym changing and despite the potentially limited observational capabilities of the adversary, as discussed in [1, 15].

Hence, there is a need for new and improved privacy protecting mechanisms that provide stronger guarantees. One promising approach is based on group signatures; however, the efficiency of those signature schemes must be substantially increased before they can be deployed in practice. In the meantime, hybrid solutions can be envisioned such as the one proposed in [5].

Moreover, attacks against privacy may happen at any layer of the communication stack. So far, most of the research efforts have focused on privacy enhancing technologies in the MAC layer and above. However, recent advances in radio fingerprinting techniques make it possible to identify a radio frequency device at the physical layer. Unfortunately, attacks at the physical layer may render protection at higher layers ineffective. Therefore, some research effort is needed to address the problem of using radio fingerprinting for tracking purposes.

Conclusions

Securing vehicular communication systems is a complex endeavor with multiple facets and subject to several unique constraints. We have systematically analyzed the problem at hand, identifying pertinent threats and models for adversaries. We have considered general security requirements and mapped those to specific VC applications. Based on a set of design principles aimed at a practical system that can readily be adopted toward deployment, we have designed a comprehensive solution, a security architecture for VC systems. We have focused on identity and credentials management, security for a variety of communication protocols, and privacy enhancing mechanisms. Furthermore, we proceeded with experimental evaluations of our mechanisms, based on simulations and prototype implementations. Our results show that with the appropriate design, secure VC systems can be practical and able to support VC applications as effectively as unsecured VC systems. Moreover, our security architecture implementation could be ported with minimal modifications to practically any platform. With the SeVeCom project reaching its conclusion, we have identified and made progress toward addressing additional research questions. This is why we believe our system can be the basis for the deployment of robust, user privacy preserving, secure VC systems.

With the SeVeCom project reaching its conclusion, we have identified and made progress towards addressing additional research questions. This is why we believe our system can be the basis for the deployment of robust, user privacy preserving, secure VC systems.

References

[1] P. Papadimitratos *et al.*, "Secure Vehicular Communications: Design and Architecture," *IEEE Commun. Mag.*, Nov. 2008.

[2] SeVeCom, "Secure Vehicular Communications: Security Architecture and Mechanisms for V2V/V2I, Delivarable 2.1," 2007–2008; http://www.sevecom.org

[3] P. Papadimitratos *et al.*, "Architecture for Secure and Private Vehicular Communications," *7th Int'l. Conf. ITS Telecommun.*, Sophia Antipolis, France, June 2007, pp. 1–6.

[4] M. Bond and R. Anderson, "API-Level Attacks on Embedded Systems," *Computer*, vol. 34, no. 10, Oct. 2001, pp. 67–75.

[5] G. Calandriello *et al.*, "Efficient and Robust Pseudonymous Authentication in VANET," *ACM VANET '07*, 2007, pp. 19–28.

[6] P. Papadimitratos *et al.*, "Impact of Vehicular Communication Security on Transportation Safety," *Proc. IEEE INFOCOM Wksp. Mobile Networking for Vehic. Environments*, Phoenix, AZ, Apr. 2008, pp. 1-6.

[7] F. Kargl *et al.*, "Secure and Efficient Beaconing for Vehicular Networks," *ACM VANET '08*, Sept. 2008.

[8] T. Leinmüller, "Car2x Communication-Challenges, Standardization and Implementation in Europe and in the US," 2007.

[9] S. Baktir *et al.*, "A State-of-the-Art Elliptic Curve Cryptographic Processor Operating in the Frequency Domain," *Mobile Networks and Apps. J.*, vol. 12, no. 4, 2007, pp. 259–70.

[10] E. Schoch *et al.*, "Communication Patterns in VANETs," *IEEE Commun. Mag.*, Nov. 2008.

[11] E. Schoch *et al.*, "On the Security of Context-Adaptive Information Dissemination," *Wiley Security and Commun. Networks J.*, vol. 1, no. 3, May 2008, pp. 205–18.

[12] M. Raya *et al.*, "On Data- Centric Trust Establishment in Ephemeral Ad Hoc Networks," *Proc. 28th IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 1238–46.

[13] P. Papadimitratos and A. Jovanovic, "Protection and Fundamental Vulnerability of GNSS," *Int'l. Wksp. Satellite and Space Commun.*, Toulouse, France, Oct. 2008.

[14] COM2REACT, "6th EU Framework Project," 2006–2007; http://www.com2react-project.org/

[15] L. Buttyan, T. Holczer, and I. Vajda, "On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs," *Proc. ESAS*, F. Stajano, *et al.*, Eds., vol. 4572, Springer Verlag, July 2007, pp. 129–41.

Biographies

Frank Kargl (frank.kargl@uni-ulm.de) received his doctorate degree in 2003 working on security in mobile ad hoc networks. His research interests center around mobile and dynamic systems, especially intervehicular 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 such as SeVeCom and PRECIOSA.

Panagiotis 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 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 an M.Sc. degree in computer science from Budapest University of Technology and Economics (BUTE) in 1995, and a Ph.D. degree from the Swiss Federal Institute of Technology, Lausanne (EPFL) in 2002. In 2003 he joined the Department of Telecommunications at BUTE, 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, VC systems, and RFID systems. More information is available at http://www.hit.bme.hu/buttyan/

Michael Müter (michael.mueter@daimler.com) graduated from the University of Technology, Aachen, Germany, in 2007 and is now working as a researcher for Daimler AG. His research interests include practical aspects of IT security, secure communication systems, and vehicular security architectures, and also theoretical aspects of reliable and dependable systems.

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. He then joined Ulm University as a researcher in the SeVeCom project, continuing to work on intervehicle communication with a special focus on robust and secure communication mechanisms.

BJÖRN WIEDERSHEIM (bjoern.wiedersheim@uni-ulm.de) studied computer science and finished his Dilpoma thesis (*Location Privacy and Tracking in VANETs*) in February 2008. After that he started to work at the Institute of Media Informatics at the University of Ulm. He is involved in the SeVeCom Project, responsible for the integration of the security components into an existing VANET stack.

TA-VINH THONG (thong@crysys.hu) received an M.Sc. degree in computer science from BUTE. Since 2008 he has been working as a Ph.D. student at CrySyS. His research interest is analyzing security systems using formal methods, especially formal analysis of security protocols. His current research activities are security API analysis and formal languages.

GIORGIO CALANDRIELLO (giorgio.calandriello@polito.it) received his doctorate degree from Politecnico di Torino in 2008 working on security of wireless networks. He is a researcher in the same institution working on security of vehicular networks, focusing on privacy and performance issues. His email is giorgio.calandriello@polito.it.

ALBERT HELD (albert.held@daimler.com) received a Dipl.- Inf. degree in computer science from Friedrich Alexander University Erlangen and a Dr.-Ing. degree in computer science from the Technical University of Dresden. In 1989 he joined the AEG Research Institute in Ulm, which later became the DaimlerBenz Research Center Ulm and is now part of Daimler Research & Development Ulm. He first worked in the field of office automation and workflow management, and then focused on IT security. Since 1993 he has worked in the areas of mobile and pervasive computing and wireless communication. He is/was project manager of several internal research projects on IT security and intelligent transportation system communications infrastructures.

ANTONIO KUNG (antonio.kung@trialog.com) has more than 25 years' experience in embedded systems. He co-founded 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 Universityand 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."