Secure and Privacy-Enhancing Vehicular Communication

Demonstration of Implementation and Operation

Petra Ardelean
EPFL, I&C-LCA 1
Lausanne, Switzerland
petra.ardelean@epfl.ch

Panagiotis (Panos) Papadimitratos
EPFL, I&C-LCA 1
Lausanne, Switzerland
panos.papadimitratos@epfl.ch
http://people.epfl.ch/panos.papadimitratos

***Abstract*—With a number of projects developing vehicular communication systems, there is rising awareness on threats and the need to introduce security and privacy-enhancing mechanisms. With recent results, in principle in agreement across different major projects, there has been little work on implementation and demonstration of security and privacy-enhancing mechanisms. The contribution of this work is exactly in this direction: we present a demonstration of our system, comprising a range of mechanisms, developed to secure vehicular communications (VC) and enhance the location privacy of the users of VC systems.**

***Keywords:* Security, Privacy, VANET, Vehicular Communications, ITS**

I. Introduction

Transportation safety and efficiency are the main driving forces for the development of vehicular communications (VC), with vehicles and roadside infrastructure units (RSUs) equipped with sensors, computers, and radios. Vehicle-to-vehicle (V2V) and infrastructure (V2I) communication include frequent beaconing of vehicle position and warnings on their condition or the environment. Typical beaconing periods for each vehicle are in the order of one beacon per 100 ms.

A strong consensus is being formed among authorities, industry, academia, and standardization bodies on the need to secure VC systems. This, to defend, for example, against an attacker that inject beacons with false information, or to collect vehicle messages and track their locations and infer sensitive user data. Major efforts to design security and privacy enhancing solutions for VC are undertaken by the SeVeCom project, the IEEE 1609 Working Group, as well as the Car-to-Car Communication Consortium (C2C-CC) Security Working Group and the CAMP and VSC projects.

Common to these efforts are: (i) a currently well-understood vehicular communication protocol stack, (ii) a Certification Authority (CA) and public key cryptography to protect V2V and V2I messages, (iii) requirements including authentication, integrity, and non-repudiation, and protection of private user information, (iv) and use of pseudonymity or pseudonymous authentication (e.g., see [1], [2]).

We briefly describe elements of such a system in Sec. II, and implementation details of the functionality that will be demonstrated in Sec. III. The demonstration story and setup are explained in Sec. IV. Our contribution is, to the best of our knowledge, the first conference demonstration of security and privacy for vehicular communications, complemented by visualizations of simulated scenarios, and a poster with experimental results.

II. Security System Elements

Requirements for secure VC systems are available in [3]. We focus primarily on the networked entities of a VC system here, rather than users. The system comprises nodes (vehicles and RSUs), and trusted third parties. Certification Authorities (CAs), instantiating the role of city, state, or country transportation authorities, are responsible for handling the long-term identities and cryptographic credentials for all network nodes in the VC system, forming a hierarchical or forest structure [2].

The basic approach to secure communication is pseudonymous authentication: each vehicle (node) is equipped with multiple certified public keys that do not reveal the node identity; the vehicle uses one pseudonym at a time and after a period it abandons it and switches to a new pseudonym. This way, messages signed under different pseudonyms cannot be linked. This is so for safety beacons (one-hop broadcast), relayed, 'floating data', and position-based routed messages; in the latter two cases, signatures from the message originator as well as the relays can be appended [5]. The pseudonyms can be obtained a priori or be reloaded, or generated by nodes on the fly [4]. Here we demonstrate the case with periodically obtained pseudonyms. In both cases, a communication transcript, under one or more pseudonyms, can be linked back to the long-term identity of the node that generated signed messages under those pseudonyms. Moreover, long- or short-term credentials can be revoked. This approach, however, is not needed for RSUs, which have a fixed identity, key, and certificate.

III. Implementation Details

Our implementation is in the C/C++ programming language, with cryptographic functions from the OpenSSL library, in Linux boxes. We follow a modular approach for an easy integration into other systems. There are three modules: Application, Security and Network and the communication between them is via sockets. While the security module is stand-alone, its implementation is transparent to the other modules, which should only be aware of the communication interfaces between them. Figure 1 illustrates this division. The

978-1-4244-1722-3/08/$25.00 ©2008 IEEE.

security module includes the following functionality: management of long- and short- term certificates and keys, generation of signed messages, verification of received signatures (including checking revocation status), and interactions with the protocol stack that relate to security and privacy (including optimizations on the security overhead [6], and communication-imposed constraints).

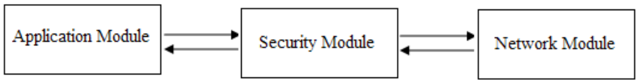


Figure 1. ***Illustration of implementation modules.***

When the application needs to send data to the network, it passes the data to the security module. Security headers are added and the newly formed packet is sent to the network module, which in turn transmits the packet across the data-link. When receiving a packet from the network, the reverse path is taken; the security module passes only validated messages to the application.

IV. Demonstration Setup and Story-Line

This demo shows how security could be implemented and integrated with VC systems, running on a few machines with a simulator used for visualization. Vehicle to vehicle, vehicle to RSU, and vehicle to CA and Pseudonym Provider communication (via an RSU) are demonstrated. Laptops communicating across a wireless interface are used. The overall setup is shown in Figure 2. Laptop A will play the role of the RSU connecting laptop B to trusted third parties, running locally and remotely. After this bootstrapping phase, laptop A will change its role and he will act as a vehicle or RSU. During all this communication, console messages will display detailed information on what happens.

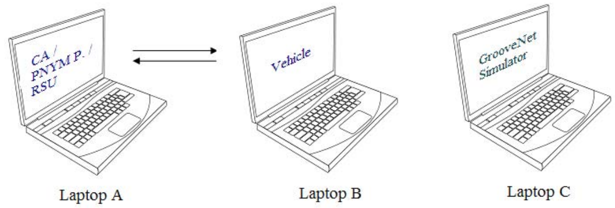
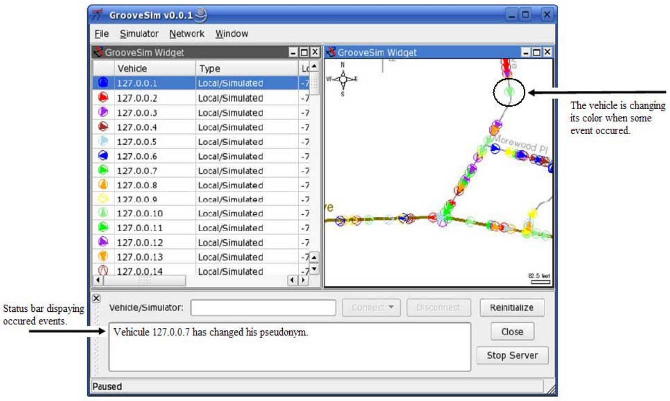


Figure 2. ***An illustration of the demo setup.***

A third Laptop C will run independently security simulations, with GrooveNet [8], [9] integrating models of attackers and security mechanisms. This enables us to visualize a number of scenarios that relate to security. We will show attack scenarios with and without the security module activated. In the latter case, attacks are not detected and vehicles consider messages valid. While, in the former vehicles detect the demonstrated attacks:

- Impersonation of a (legitimate) vehicle, to inject false information
- Violation of access control policies; injection of messages the vehicle is not authorized to send (e.g. acting as a RSU, a bus, or a police car)
- Replay and neighborhood discovery attacks [7]
- Vehicle tracing by a subset of rogue RSUs

When an event occurs, a message is displayed in the status bar of the window and cars involved in the event change colors. Figure 3 shows a screenshot of the simulator window.



	Vehicle	Type	Lc
🔵	127.0.0.1	Local/Simulated	-7
🔴	127.0.0.2	Local/Simulated	-7
🟣	127.0.0.3	Local/Simulated	-7
🔴	127.0.0.4	Local/Simulated	-7
🔵	127.0.0.5	Local/Simulated	-7
🔵	127.0.0.6	Local/Simulated	-7
🟢	127.0.0.7	Local/Simulated	-7
🟠	127.0.0.8	Local/Simulated	-7
🟡	127.0.0.9	Local/Simulated	-7
🟢	127.0.0.10	Local/Simulated	-7
🟢	127.0.0.11	Local/Simulated	-7
🟣	127.0.0.12	Local/Simulated	-7
🟠	127.0.0.13	Local/Simulated	-7
🔴	127.0.0.14	Local/Simulated	-7

Vehicule 127.0.0.7 has changed his pseudonym.

The vehicle is changing its color when some event occured.

Status bar dispaying occured events.

Figure 3. ***Screenshot from visualization of security and privacy-related events with the GrooveNet simulator.***

Acknowledgment

We are grateful to our colleagues at (i) EPFL and notably the LCA1 unit and J.-P. Hubaux, (ii) the SeVeCom consortium, (iii) the C2C-CC Security Working Group, (iv) the eSafety eSecurity Working Group and (v) the IEEE 1609.2 Working Group, for their support, collaboration, and valuable feedback.

References

[1] SEVECOM Project, "Security architecture and mechanisms for V2V / V2I, Deliverable 2.1," URL: http://www.sevecom.org

[2] P. Papadimitratos, L. Buttyan, J-P. Hubaux, F. Kargl, A. Kung, and M. Raya, "Architecture for secure and private vehicular communications," 7th International Conference on ITS Telecommunications, Sophia Antipolis, France, June 2007

[3] P. Papadimitratos, V. Gligor, and J.-P. Hubaux, "Securing vehicular communications - assumptions, requirements, and principles," Workshop on Embedded Security in Cars (ESCAR) 2006, Berlin, Germany, November 2006

[4] G. Calandriello, P. Papadimitratos, A. Lioy, and J.-P. Hubaux, "Efficient and robust pseudonymous authentication in VANET," The Fourth ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2007), in conjunction with ACM MobiCom 2007, Montréal, QC, Canada, September 2007

[5] C. Harsch, A. Festag, and P. Papadimitratos, "Secure position-based routing for VANETs," IEEE VTC2007-Fall Conference, Baltimore, MD, October 2007

[6] P. Papadimitratos, G. Calandriello, J.-P. Hubaux, and A. Lioy, "Impact of vehicular communication security on transportation safety," IEEE Infocom MOVE 2008, Phoenix, Arizona, USA, April 2008

[7] M. Poturalski, P. Papadimitratos, and J-P. Hubaux, "Secure neighbor discovery in wireless networks: formal investigation of possibility," ACM ASIACCS 2008, Tokyo, Japan, March 2008

[8] R. Mangharam, D. Weller, R. Rajkumar, P. Mudalige, and F. Bai, "GrooveNet: A hybrid simulator for vehicle-to-vehicle networks"

[9] http://www.seas.upenn.edu/~rahulm/Research/GrooveNet