Using TPMs to Secure Vehicular Ad-hoc Networks (VANETs)

Gilles Guette[1] and Ciarán Bryce[2]

IRISA
Campus de Beaulieu, 35042 Rennes CEDEX, France
gilles.guette@univ-rennes1.fr, Ciaran.Bryce@inria.fr

Abstract. Vehicular Ad hoc Networks are the focus of increased attention by vehicle manufacturers. However, their deployment requires that security issues be resolved, particularly since they rely on wireless communication, and rogue vehicles can roam with contaminated software. This paper examines security threats to VANETs and argues that a security architecture built around TPMs can provide a satisfactory solution.

1 Introduction

Over the last few years, Vehicular Ad hoc Networks (VANETs) have gained much attention within the automobile and research worlds. One reason is the interest in a growing number of applications designed for passenger safety – such as emergency braking, traffic jam detection and cooperative driving – as well as in applications aiming at the comfort of passengers, such as games, chat-rooms and vehicle data-sharing (e.g., CarTorrent [1]).

VANETs are highly dynamic *ad hoc* networks of devices with very restricted access to a network infrastructure. Moreover, if base stations are sparsely deployed along the road, access is also of short duration due to vehicle speed. Since on-board applications need to exchange data, the communication security problem must be addressed. The absence of a permanently present infrastructure means that a decentralized security architecture is required. Given the safety critical nature of some VANET applications, the security architecture must imperatively prevent a malicious person from successfully launching an attack intending to provoke collisions between vehicles.

This paper examines some of the security requirements for VANETs for two selected applications – platoons and event reporting. The security analysis insists on the need for pseudonymity, trustworthy information exchange and a fail-safe mode where doubts over the trustworthiness of information can be conveyed to the vehicle (driver). We contend that these requirements can be met in a scalable way by a security architecture built around the emerging Trusted Platform Module (TPM) [2] specification. This is partly because use of the TPM makes it easier to verify that correct functioning of software on a vehicle, and the distribution of keys for TPM operation can be accommodated by current vehicle registration and maintenance practices.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 presents two example VANET applications along with their security requirements. The TPM is presented in Section 4, and the TPM based security solution is outlined in Section 5. Section 6 concludes the paper.

2 Related Work

In a wireless Vehicular Ad hoc Network, as data is broadcast over a shared communication media, it is simple for a malicious node to intercept or modify data, or to inject erroneous data. A data injection can provoke collisions in a vehicular platoon [3]. The open nature of a VANET thus renders communication security a great challenge [4,5,6,7].

One approach to VANET security has been to adopt a VANET PKI (VPKI) [8,9] that allows vehicles to securely communicate among themselves. Base stations placed along the road provide support for the infrastructure, notably for key distribution and revocation.

VPKI solutions address privacy using an *anonymous key set* and a *key changing algorithm* to avoid the possibility of car tracking. Without key changing, a vehicle would use the same public key to sign all of its messages. It would thus be easier for an eavesdropper on the network to correlate the vehicle's positions with the public key holder.

VPKIs are promising for VANET applications. However, the PKI deployment is a large-scale and potentially costly procedure since it requires large-scale testing after deployment to ensure operation under real-world VANET conditions. Further, the solution really only aims at ensuring authentication (of pseudomyns). As we argue in the next section, a security infrastructure must be aimed at establishing the authenticity of message contents for safety and security.

Some other papers address the problem of privacy [10,11,12] in VANET with the help of infrastructures (base station and certification authorities) and pseudonym use. [12] deals with the challenges encountered when applying anonymity to a VANET communication system and proposes a framework for pseudonymity support. A study of the impact of pseudonym changes on geographic routing in VANETs is made in [13]. All of these papers underline that supporting pseudonymity requires changing other *identifiers* of the protocol stack, such as IP or MAC addresses.

3 Use Case: Cooperative Driving

In this section, we present two cooperative driving applications for Vehicular Ad hoc Networks. This is the subject of Section 3.1. Section 3.2 then presents the security threat model for these applications, and Section 3.3 finishes with a list of desired security properties.

In the context of this paper, vehicles are nodes of an *ad hoc* network, and we use the terms *node*, *car* and *vehicle* interchangeably. Each car has wireless networking capabilities (e.g., ad hoc WLAN) and possesses a GPS device for positioning itself. A vehicle may have further sensor devices, e.g., for sensing the weather conditions. The set of sensors maintained by a vehicle is termed its *configuration*.

Note, we do not expect each car circulating on the road to be part of a VANET. Rather, VANET functionality will be something progressively introduced into new cars. Even then, there will always be old model cars as well as foreign vehicles on the roads. Thus, the cooperative driving use cases do not rely on all cars being VANET nodes.

3.1 Description

Each node has embedded sensors to detect environment information. While the sensor configuration may differ from one node to another, all nodes of the VANET use wireless communication to broadcast and share information obtained via their sensors about the state of the traffic – traffic jams, road fluidity, obstacles, weather conditions, *etc.* As suggested in Figure 1, information sharing may help to avoid accidents by enabling drivers to adapt their behavior based on pertinent safety information from vehicles driving in the opposite direction.

Another cooperative driving application, based on inter-vehicle information sharing information, is vehicular platoons [3], *c.f.*, Figure 2. A platoon reduces the distance between vehicles, and this has the economic and ecological advantage of reducing fuel consumption. The application embedded in the vehicle manages the distance between a vehicle and its predecessor and successor vehicles, and manages variations in speed.

Both of these applications rely on the vehicles' configuration returning accurate readings. For instance, the GPS module of a given vehicle can

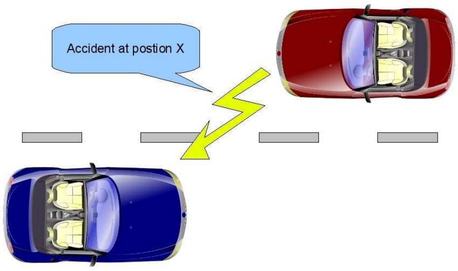

Fig. 1. Data transmission

make an error in positioning, or its internal clock may be erroneous, thus indicating an event that is more recent than its message suggests. Further, every car is different – the time it takes to accelerate or decelerate is different for each car and this has an important impact in the platoon application. It is therefore important for the vehicle's configuration to be up-to-date with respect to device and sensor characteristics: the vehicle must continuously measure its configuration so that other nodes can interpret messages received from it with respect to sensor accuracy.

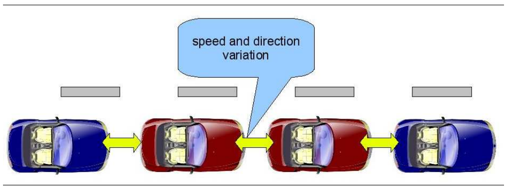

Fig. 2. Vehicle platoons

3.2 Threat Model

As mentioned, the reception of traffic information can modify the behavior of a driver. When a message announces an obstacle on the road or fog a few kilometers ahead, the control screen of the vehicle or a digital voice alerts the driver. There is a high probability that the driver slows down.

The reduction of distances between vehicles increases the risk of collisions if an attacker can send wrong information to a vehicle at the wrong moment. We need to avoid such a possibility because this kind of attack can have catastrophic consequences. Generally, the security architecture has to deal with the following attacks.

The Sybil Attack The Sybil attack was first described and formalized by Douceur in [14]. It consists of a node sending multiple messages, with each message containing a different, fabricated, source identity. Thus, the attacker appears in the network as a large number of different nodes. Applications of the Sybil attack to Vehicular Ad-Hoc Networks are discussed in [3,15] and show the importance of Sybil node detection in VANETs. A possible goal of a Sybil attack by an attacker is to give the illusion to other cars that there is a traffic jam and thus encourage other vehicles to leave the road to the attacker's benefit. Nevertheless, this attack may be more dangerous, targeting directly human life by trying to provoke collisions [3].

One important result shown in [14] is that without a logically centralized authority, Sybil attacks are always possible (*i.e.* may remain undetected) except under the extreme and unrealistic assumption of resource parity and coordination among entities.

Node Impersonation Drivers are legally responsible for their actions behind the wheel. In the event of an accident or a driving offense, there is a need for the police to associate the implicated vehicle with its driver. This is currently possible thanks to databases of driving license plates. In a VANET, this can be easily accomplished by giving a unique identifier to every vehicle. In case of an accident provoked by wrong information sent by a vehicle, the message can be verified and its identifier controlled. The police may then bind the identifier with the driver's identity.

This identifier must be protected so that an attacker cannot masquerade with a fabricated or some other car's identity. At the same time, for privacy reasons, it must not be possible for someone to deduce the driver's identity from the vehicle's VANET identifier.

Sending False Information An attacker may want to send wrong or forged data to other vehicles to provoke collisions, to free the road or for some other goal. This threat against the vehicle may be mitigated by the fact that there exists a way to know the identifier of the sender of a message. Nonetheless, the security mechanism must integrate ways to estimate the truthfulness of information.

Car Tracking Driver privacy is a concept that must be integrated into the security solution. Drivers may wish to preserve their anonymity even though the use of unique identifiers allows vehicles to be tracked. Nevertheless, in [6] the authors underline an important fact: today vehicle are only partially anonymous. Drivers implicitly surrender a portion of their privacy since each vehicle has a publicly displayed license plate that uniquely identifies it. It might not be difficult to link a license plate to the driver's identity.

The use of wireless communication does not add a new problem threatening the driver's privacy. Nevertheless, as data is broadcasted over a potentially long range, it becomes easier to collect data. Moreover, if base stations are deployed along the road, data might be collected by a third party with a commercial aim. Solutions based on the use of pseudonyms are presented in [12,10,11]

3.3 Basic Security Properties

Regarding the different threats exposed in the previous section, we can define basic properties that a security solution must provide.

Property 1. A node must have a unique identifier. This identifier may be associated with a set of pseudonyms, but in this case an authority must have the possibility of linking a given pseudonym to its associated unique identifier.

Property 2. To avoid modification of a given message or a wrongful claim of identity in a message, each message must be authenticated with regards to a vehicle identifier, and the integrity of this message must be ensured.

To engage the liability of a driver having caused an accident, non-repudiation must also be provided by the security solution. Nevertheless, this security property is implicitly provided by the combination of properties 1 and property 2 – if a message containing the unique identifier of its sender cannot be modified, then non-repudiation is effectively provided.

Property 3. The trustfulness of message contents must be verifiable.

This is a stronger property than before. In effect, it entails being able to challenge a vehicle for it to prove that its configuration readings are correct. In effect, the security infrastructure is more linked to demonstrating correct functionality rather than identity.

One way to avoiding false information exchange is to authenticate the application, rather than the vehicle, sending the information. If we can prove that information have been sent by a cooperative driving application and that this application has no been hacked, we can ensure that this information is not voluntarily wrong.

Avoiding information that is involuntary wrong, e.g., due to erroneous sensor readings, is achieved by challenging vehicles for readings while in the same geographical vicinity. In this case, the readings of the challenger and the challenged vehicle should not significantly diverge. If they do, then either vehicle has an error and should avoid a platoon with other vehicles.

4 The Trusted Platform Module

Implementing the security properties that we presented in Section 3 requires that a vehicle be able to establish trust in another vehicle, even though that vehicle is under the complete control of an unknown, and therefore untrusted, driver. The solution thus requires the use of secure hardware. An example of a general purpose hardware chip designed for secure computing is the *Trusted Platform Module* (TPM) [2] which can be integrated into any device. TPMs are now shipped with PCs; 200 million TPM-enabled PCs have been shipped by the end of 2007.

A TPM is a piece of hardware, requiring a software infrastructure, that is able to protect and store data in shielded locations. A TPM has also cryptographic capabilities such as a SHA1 engine, an RSA engine and a random number generator. Figure 3, taken from [2], illustrates the main components of a TPM.

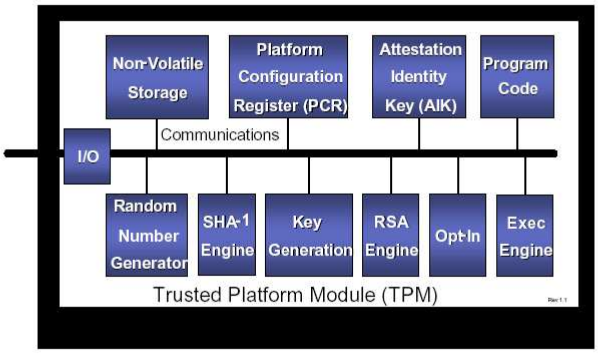

Fig. 3. Architecture of a TPM

A trusted platform must provide three basic features: protected capabilities, integrity measurement and integrity reporting. Integrity measurement and reporting mean that at the first boot of a platform, the TPM is able to store a fingerprint of application and environment variables in a specific shielded location called a Platform Configuration Register (PCR). In principle, any change that an attacker tries to make to the application will lead to a difference between the original fingerprint of the application and the fingerprint of the hacked application, thus allowing other devices to detect that a hacked application is being run.

The data used to take fingerprints are stored by the platform in a Stored Measurement Log (SML); only the digests of this data are stored in PCRs. During a challenge, the challenger requests to see specific PCR values. Then, an agent on the platform collects the SML entries and receives the PCR values from the TPM. The challenged TPM signs the PCR values with an *attestation identity key* (AiK). The platform agent collects certificates, or credentials in TPM parlance, the signed PCR values, SML entries and returns these to the challenger. Finally, the challenger verifies all the received elements. This procedure is known as an *attestation* protocol.

The credentials involved in attestation serve to demonstrate that the TPM is operating correctly and that the AiK was generated by a valid TPM. There are several keys and credentials, the most important being:

- Each TPM has a unique master key called an Endorsement Key (EK). This is a pair of RSA keys with a minimum size of 2048 bits. Storing this key inside the TPM ensures its security. The public part of the EK is available in the Endorsement Credential. This credential is available outside the TPM itself. The EK is generated by the TPM constructor.
- A Platform Credential is created and signed by the platform provider in which the TPM is integrated and identifies the platform. Generally the platform provider, or some entity that he trusts, will test the TPM and issue a Conformance Credential for the TPM. The conformance credential proves that the TPM has passed the different phases of evaluation.
- A TPM can generate AiKs for attestation protocols. However, credentials must also be issued for these keys that certify that the TPM that generated the key is valid. The TPM specification describes a protocol where a trusted party known as a *privacy CA* can generate AiK credentials for TPMs. The advantage of this is that AiKs credentials need not disclose platform identity in an attestation protocol.

The TPM embedded also offers the possibility of creating and storing encryption keys for data. This feature can be used by vehicles to store driver data securely. For this reason, and also because there is a mapping between vehicle registration and review organizations in practice allowing the appointment of privacy CAs, means that the TPM is a good solution on which to base a security architecture. This is the subject of the next section.

5 Exploiting the TPM for Use Case Security

The security model works at two levels. The basic level permits a *trusted channel* to be established between any two vehicles. This means that the two vehicles are satisfied that each is running an untampered version of the security software, and that no intentional data attack or Sybil attack is being attempted. The second level aims at *information verification.* It builds on trusted channels to offer means to ensure that a vehicle's configuration does not contain erroneous readings.

Implementing trusted channels relies directly on the TPM's attestation mechanism. A vehicle can trust another if the latter can demonstrate that its software has not been tampered with, and the source of the software can be verified. The issue in deploying a TPM on VANET nodes is to assign roles to the actors in the TPM protocols. We assume the following:

- Car manufacturers sign the platform credentials for their vehicles. It is logical to assume that a manufacturer takes responsibility for all embedded devices on their vehicles. Further, manufacturers are relatively few in number and are "well-known" in the sense that certificates signed by these principals should be recognizable to all vehicles and automobile authorities.
- Automobile authorities are responsible for organizing technical reviews. In most countries, car owners are obliged to submit their car to a technical review every 2 to 3 years. A car that fails the technical review cannot be driven on the road. Automobile authorities are thus well-known principals that can act as privacy CAs that can sign AiK credentials.

The TPM provides a means to securely attribute a vehicle identifier. This can be signed by an automobile authority, thus ensuring Security Property 1 of Section 3. The attestation protocol used when vehicles exchange information then ensures Security Property 2.

The second level of security, information verification, is based on three simple procedures. These guarantee Security Property 3.

1. *Auto-measuring.* A vehicle's software maintains data on the vehicle's acceleration and deceleration capabilities, as well as related data such as tire denseness (which embedded devices are now able to measure). These values evolve so the vehicle continuously updates them. These values are obviously important for the platoon scenario where neighboring cars need to agree on minimal distances.
2. *Challenge-response* protocol. This procedure is needed to detect unintentional errors in information transmitted by a vehicle that are due to permanent errors in the sensor of the car. Cars that are close together should possess the same readings for many information types, e.g., temperature, time, location, luminosity. The goal is thus to permit a car to challenge another with respect to any of these readings.
3. *Technical review.* Technical reviews – organized by automobile authorities – for cars with VANET functionality must include reviews of the correct functioning of all sensor devices. Further, we expect that any changes that need to be made to the application software is made at this moment. This is important since the TPM can only be used to help verify that the software on a platform has not been tampered with; in no way does this guarantee that absence of security flaws or bugs in the software itself.

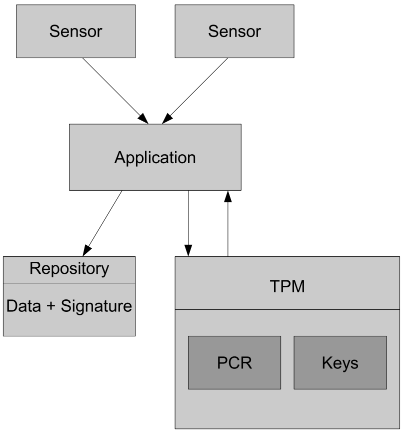

Fig. 4. The embedded architecture

The three procedures help to detect and isolate permanent errors in readings. Obviously, intermittent errors are not necessarily treated, and we will look more into this issue in future work. However, we note that these errors are especially a problem for the information exchange scenario of Figure 1 and less so for the platoon scenario (of Figure 2). The latter is more safety critical.

Figure 4 shows the different components of the embedded architecture and the data flow. For instance, for auto-measuring, sensors embedded in vehicle give results of their measures to the application. Then the application asks the TPM to sign the data, the TPM checks the PCR value associated with this application and signs data provided by the application. Then the application can store this data in a dedicated repository.

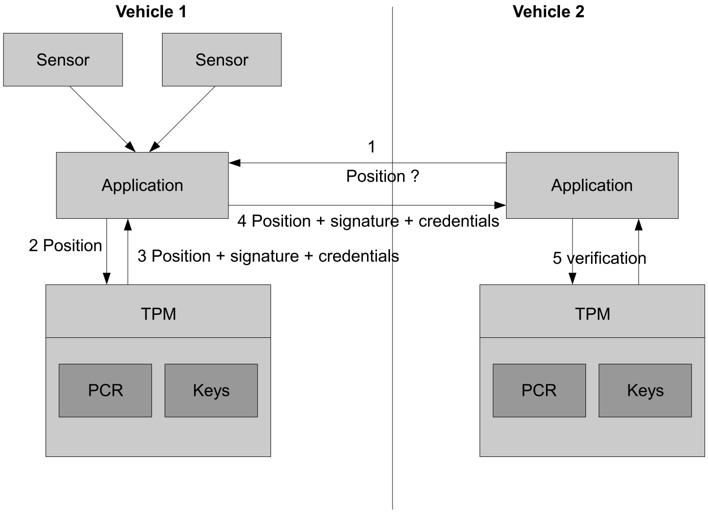

Fig. 5. The challenge-response protocol

The details for challenging another vehicle in order to detect unintentional errors is given in Figure 5. The challenger sends a query about data it can verify, the current position in the exemple. Then the challenged vehicle collects the appropriate data, gives this data to its TPM. The TPM checks the PCR values associated with this application and signs data. The application sends to the challenger the signed data and associated credential. The challenger verifies the signature and then can compare the given position to its own current position to detect misconfiguration of the positioning unit of the challenged vehicle.

6 Conclusion and Future Work

In this paper, we have presented the benefit provided by the TPM architecture in Vehicular Ad hoc Networks. We described an application of cooperative driving and its associated threat model. We claim that an embedded TPM inside vehicles can greatly increase the security of wireless communication in this kind of network, and serves as a basis for detecting both intentional and accidental attacks. We are currently working on improvements of our model, notably to the way that updates to application code and embedded certificates are handled.

References

1. Lee, K.C., hoon Lee, S., Cheung, R., Lee, U., Gerla, M.: First Experience with CarTorrent in a Real Vehicular Ad Hoc Network Testbed. In: Mobile Networking for Vehicular Environments. (2007) 109–114
2. Trusted Computing Group: TPM main specification. Main Specification Version 1.2 rev. 85, Trusted Computing Group (February 2005)
3. Blum, J., Eskandarian, A.: The Threat of Intelligent Collisions. IT Professional **6**(1) (January-February 2004) 24–29
4. Zarki, M.E., Mehrotra, S., Tsudik, G., Venkatasubramanian, N.: Security Issues in a Future Vehicular Network. In: European Wireless. (2002)
5. Hubaux, J., Čapkun, S., Luo, J.: The Security and Privacy of Smart Vehicles. IEEE Security and Privacy **2**(3) (May-June 2004) 49–55
6. Parno, B., Perrig, A.: Challenges in Securing Vehicular Networks. In: Fourth Workshop on Hot Topics in Networks. (2005)
7. Aijaz, A., Bochow, B., Dötzer, F., Festag, A., Gerlach, M., Kroh, R., Leinmüller, T.: Attacks on Inter-Vehicle Communication Systems - An Analysis. In: 3rd International Workshop on Intelligent Transportation. (2006)
8. Raya, M., Hubaux, J.: The Security of Vehicular Ad Hoc Networks. In: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks. (2005) 11–21
9. Raya, M., Papadimitratos, P., Hubaux, J.: Securing Vehicular Communications. IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications **13**(5) (2006) 8–15
10. Dötzer, F.: Privacy Issues in Vehicular Ad Hoc Network. In: Workshop on Privacy Enhancing Technologies. (2005) 197–209
11. Gerlach, M., Festag, A., Leinmller, T., Goldacker, G., Harsch, C.: Security Architecture for Vehicular Communication. In: Workshop on Intelligent Transportation. (2007)
12. Fonseca, E., Festag, A., Baldessari, R., Aguiar, R.: Support of Anonymity in VANETs - Putting Pseudonymity into Practice. In: IEEE Wireless Communications and Networking Conference. (2007)

13. Schoch, E., Kargl, F., Leinmüller, T., Schlott, S., Papadimitratos, P.: Impact of pseudonym changes on geographic routing in vanets. In: European Workshop on Security in Ad-hoc and Sensor Networks. (2006) 43–57
14. Douceur, J.: The Sybil Attack. In: First International Workshop on Peer-to-Peer Systems. (March 2002) 251–260
15. Raya, M., Hubaux, J.: Securing Vehicular Ad Hoc Networks. Journal of Computer Security, Special Issue on Security of Ad Hoc and Sensor Networks **15**(1) (2007) 39–68