Creating a Secure Infrastructure for Wireless Diagnostics and Software Updates in Vehicles

Dennis K. Nilsson, Ulf E. Larson, and Erland Jonsson

Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
{dennis.nilsson,ulf.larson,erland.jonsson}@chalmers.se

Abstract. A set of guidelines for creating a secure infrastructure for wireless diagnostics and software updates in vehicles is presented. The guidelines are derived from a risk assessment for a wireless infrastructure. From the outcome of the risk assessment, a set of security requirements to counter the identified security risks were developed. The security requirements can be viewed as guidelines to support a secure implementation of the wireless infrastructure. Moreover, we discuss the importance of defining security policies.

Keywords: Infrastructure, vehicle, wireless, security, guidelines, policies.

1 Introduction

This paper presents guidelines for creating a secure infrastructure involving wireless communication for performing diagnostics and software updates in vehicles. It is assumed that both wireless diagnostics and software updates use the same communication channel and security principles. We assume that the security requirements for the communication channel are the same for both wireless updates and wireless diagnostics.

Today, vehicles contain a number of *electronic control units* (ECU). These units are responsible for various functionality in the vehicle, ranging from small tasks such as opening a window or unlocking a door to more advanced functionality such as automatic brake systems and collision warning systems [1]. Each ECU runs its own specific and independent software. As with all software, new improved versions are created to remedy bugs and add new functionality. As new releases of software are available, the customer can update the software for the corresponding ECUs by visiting an authorized service station. The service station employee sets up a wired connection to the vehicle to update the software. The new software is downloaded and flashed to the ROM of the particular ECU, overwriting the old software. In addition to software updates, diagnostics can be performed on the ECUs to detect errors or to determine the cause of malfunctions. For example, if the head lights do not turn on, diagnostics can be performed at a service station to find the cause of the problem (e.g., a faulty fuse). Diagnostics is also performed in test environments to test functionality

M.D. Harrison and M.-A. Sujan (Eds.): SAFECOMP 2008, LNCS 5219, pp. 207–220, 2008.

(e.g., lock and unlock the passenger door) and find errors in an early phase before the software is released. These procedures which today require physical access to set up a wired connection, may be inconvenient for the customer as well as for the service station.

Thus, software updates via a wireless communication channel emerges as a promising possibility. The benefits are several, including minimal customer inconvenience, mass updates, and faster updates. In addition, it allows improved testing and reduced time from fault to action [2].

We have analyzed the wired diagnostics and software update procedures, and assessed the security risks associated with a wireless infrastructure. As a result of the risk assessment, we provide a set of security requirements that can be viewed as guidelines for creating a secure infrastructure for wireless diagnostics and software updates. This paper is a revised version of a more extensive work [3].

The paper is outlined as follows. Section 2 discusses related research in this area. In Section 3, we identify the attacker model, and define desired security properties and assumptions in the wireless infrastructure. In Section 4, we assess the risks for a wireless infrastructure. Section 5 presents the guidelines for creating an infrastructure for wireless diagnostics and software updates. In Section 6, we discuss importance of defining security policies. Section 7 provides possible future work directions, and Section 8 concludes the paper.

2 Related Work

The research in this area is often very specific and usually targets only one part of the infrastructure: the communication part. For example, Mahmud et al. present an architecture for secure software upload to vehicles using wireless communication links, where the focus is on the communication links [4]. In their solution, a set of authentication keys are installed in a vehicle at production time. A central server has the same set of authentication keys. Once authentication has been performed, the central server issues a symmetric session key to the vehicle. The symmetric session key is then used for secure communication with the vehicle during that session.

A discussion on securing vehicular communications is presented in [5]. The discussion describes challenges and vulnerabilities in vehicular communications. However, the vehicular communications only involve communications between vehicles, and not communications with third parties for updating software. A security architecture using secure hardware, vehicular public key infrastructure and new methods for certificate revocation is also presented.

A comparison of different flashing methods for software updates is described in [6]. The physical connection reflashing method is compared to software updates over the air at a service station and at the customer location. The incentives and challenges for updating software over the air are presented; however, no proposals for solving the mentioned challenges are given.

A proposal of using multicasting to update the software in a large number of vehicles is presented by Miucic et al. [7]. The security issues are carefully

discussed, and the idea of a decentralized key management system, where multicast session keys are generated and distributed to group members is presented. By using an encryption key, shared by authorized members only, the security of the multicast communication, i.e., the confidentiality and integrity of the transmitted data and the authenticity of the group members, is achieved. Digital certificates are used to provide source authenticity and integrity of the multicast data. However, in this paper, we assume that vehicles use different software and software versions, which makes multicast data not useful for our scenario. We need to establish individually secure end-to-end communication.

There also exist several patents [8,9,10,11] in the area of the wireless diagnostics and software updates but the descriptions are often very high-level and do not contain any security-relevant details.

There have been substantial more work in this area; however, the work typically focuses on one aspect of the infrastructure while we take on a broader perspective of providing *guidelines for creating a secure infrastructure for wireless diagnostics and software updates in vehicles.*

3 Background

In this section, we describe an attacker model that is specific for the wireless infrastructure. We also present the desired security properties and assumptions we make about the wireless infrastructure. In addition, we discuss the hardware constraints that exist in the vehicular environment.

3.1 Attacker Model

We categorize our attacker as either an *insider* or an *outsider* [12,13]. An insider is an authorized member of a system, in this case the infrastructure. Basically, an insider can perform any action the authorized user can and, in addition, can mount attacks from inside the system.

An outsider is considered an intruder to the system and can only mount attacks from outside the system. For example, an outsider attacker can attack the wireless communication link. In order to address this problem, we adopt the Dolev-Yao attacker model [14], where an attacker can eavesdrop, intercept, modify or inject messages into the communication link. Moreover, after a successful intrusion, an attacker can gain access to the internal network of the vehicle or the portal, and thus execute attacks as an insider.

3.2 Desired Security Properties

In this section, we list the desired security properties for wireless diagnostics and software updates in vehicles.

Confidentiality
The software to be installed in the ECUs is proprietary and should be kept confidential. This includes the storage and the transmission of software binaries.

The transmitted diagnostics requests and replies as well as the stored diagnostics data should also be kept confidential.

Integrity
The software to be installed in the ECUs is used for controlling safety and security-critical features and needs to be protected against modification. The data integrity of the software must be verified such that a vehicle can assure that the correct software has been received.

Authentication
The communication between the portal and the vehicle needs to be authenticated. Mutual authentication is required to prevent impersonation of either portal or vehicle. Moreover, data authentication is needed such that the vehicle can verify that the received software comes from a trusted source.

Freshness
To protect against replay attacks, for example, replaying a diagnostics request to turn off the head lights, the protocol must ensure that the messages are fresh.

Resilience to lost packets
Since wireless communication is susceptible to packet loss, the infrastructure must be designed to handle lost packets in a graceful and secure way. The communication link must also be specifically protected against denial-of-service (DoS) attacks to preserve the availability of the link.

3.3 Assumptions about the Wireless Infrastructure

We assume that a centralized architecture is used, since the proprietary software and secret cryptographic keys are stored at a central location, which we denote the *portal*. The portal communicates with a large number of vehicles, and each vehicle is treated individually in terms of software and cryptographic keys. In other words, each vehicle has its own set of installed software and keys. Therefore, the portal must store the state (current software versions and keys) corresponding to each vehicle. We assume that the portal consists of high-computational devices with large storage areas, and therefore storing and accessing this data in the portal is *not* a problem.

Furthermore, since diagnostics and software updates are performed at an infrequent basis per vehicle, we assume that the communication, computation and memory overhead at the portal for each instance is insignificant. In other words, we assume that the portal will *not* be a bottleneck for wireless diagnostics and software updates, even for a large number of vehicles.

We further assume that necessary cryptographic keys (e.g., authentication keys) are distributed offline and installed in the vehicles during manufacturing. Therefore, key management is *not* an issue, since the portal and the vehicles already have established keys when the vehicles are deployed in the network. Moreover, since we assume that the established keys in the vehicles will be used for the rest of their lifetime, rekeying is *not* considered.

3.4 Limited CPU Processing Power and Memory Size

Most ECUs in the vehicle have very limited CPU processing power and memory size. This limits the possibility to use heavy cryptographic algorithms in the encryption and authentication procedures. Also, the downloaded software binaries might not fit in the ECU RAM[1] meaning that the binaries must be temporarily stored somewhere else. Issues that need to be resolved include the storage of encryption keys for the temporarily stored software binaries. Moreover, incorporating a firewall, IDS or logging utility in the vehicle also requires careful consideration with respect to the limited hardware resources.

4 Assessing Security Risks for a Wireless Infrastructure

A traditional wired infrastructure, containing the three regions *portal*, *communication link*, and *vehicle* [6], is illustrated in Fig. 1 and can be described as follows. The portal is communicating with a vehicle over a wired connection. For software updates the portal accesses data (the software to be installed in the ECU) in the internal portal network and sends the data to the vehicle over the wire. Once received in the vehicle, the data is routed through the in-vehicle network and installed in an ECU inside the vehicle. The procedure is similar for diagnostics requests.

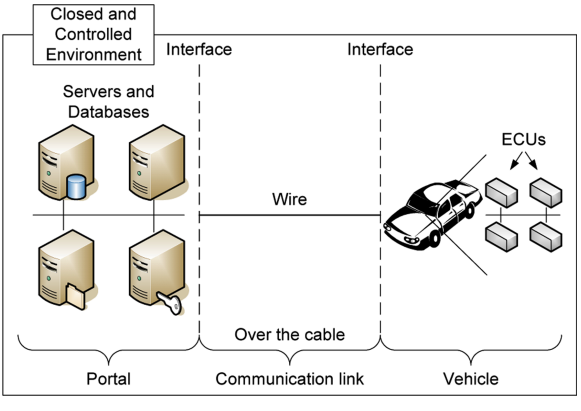

Fig. 1. Infrastructure for wired diagnostics and software updates

4.1 Risk Assessment for a Wireless Infrastructure

For the wired scenario, the procedures for diagnostics and software updates require physical access to the vehicle to connect it to the wire. Moreover, the portal, the wire, and the vehicle are in a closed and controlled environment under immediate supervision. This scenario can therefore be considered as relatively secure against attacks, especially outsider attacks, but when the same

[1] The software binary is downloaded to the RAM and then flashed to the ROM. The ROM could be larger than the RAM.

procedures are performed in a wireless infrastructure the scenario drastically changes. We therefore perform a risk assessment using the attacker model, desired security properties and assumptions, described in Section 3, as a basis, and list the assessed security risks of attacks for each of the three regions in the following paragraphs. We use traditional computer and network attacks [15] as a basis to develop the security risks in our scenario. In addition, we list the risks of consequences as a result of such attacks.

4.2 Portal Security Risks

The portal is still in a controlled environment but by setting up wireless communication links to the outside, the environment is no longer closed since an entry point[2] to the portal is introduced. The following risks are identified.

1. **Impersonation**
   The risk for an impersonation attack increases. For a wire it is possible to know that the vehicle is connected to the portal by physically following the wire but for wireless communication this is not possible. An attacker can impersonate the portal and establish a connection to a vehicle.
2. **Intrusion**
   The entry point to the portal also poses a security risk. A weakness in the portal could allow an intrusion, which in turn could potentially allow the outsider attacker equal access to that of an insider. An insider can access sensitive and proprietary data and execute more serious attacks.

4.3 Communication Link Security Risks

The communication link is no longer in a controlled and closed environment. The wire is replaced with communication over the Internet and over-the-air. The following risks are identified.

3. **Traffic Manipulation**
   The risk that an attacker can inject or modify packets in the communication link is increased, especially due to the added exposure caused by the wireless communication. This attack could cause diagnostics to perform actions that they were not originally intended to perform (e.g., unlock the door instead of checking if the door was locked). An attacker can also replay, for example, a diagnostics request to unlock the door.

4.4 Vehicle Security Risks

The vehicle is no longer in a controlled and closed environment, and immediate supervision may not be possible. The following risks are identified.

[2] We define an entry point as a communication interface that allows entry to an internal network.

4. **Impersonation**
   The risk for an impersonation attack is increased. For wired communication, it is possible to physically follow a wire to know which vehicle is connected but for wireless communication this is not feasible. An attacker can impersonate a vehicle and set up communication links with the portal.
5. **Intrusion**
   The wireless interface to the vehicle also introduces an entry point. A weakness in, for example, the authentication procedure in the vehicle could allow intrusions, which could potentially allow an outsider attacker equal access to that of an insider. An insider can access sensitive and proprietary data and execute more advanced attacks.

4.5 Risks of Consequences

If the attacks on the portal, communication link, and vehicle are successful, the consequences could be disastrous.

6. **Execution of Arbitrary Code**
   With both wireless diagnostics and software updates, it is possible to affect the behavior of the ECUs. Thus, as a result of a successful impersonation of the portal or an intrusion attack to the portal, an attacker can issue diagnostics requests or software that execute in the vehicle which believes the requests or software originated from the real portal. Thus, an attacker can run arbitrary code on the vehicle. A rational attacker can read confidential data from the vehicle or, for example, unlock the driver door. A malicious attacker can cause damage by, for example, disabling the brakes in the vehicle. Furthermore, an attacker who has access to the internal portal network can perform attacks as an insider. In addition, a successful intrusion attack to the vehicle could allow an attacker to update the ECUs with modified versions of software, where the attacker can control the functionality of the ECUs. A rational attacker can update the software in ECUs with performance-enhanced versions of the software. A malicious attacker, on the other hand, can update the ECUs with malicious versions of the software that can cause damage to the vehicle or injury to a person (e.g., triggering the airbag remotely when a person is sitting in the seat).
7. **Disclosure of Information**
   A successful intrusion to the portal may allow an attacker to learn private information about customers and access proprietary software. Moreover, a successful impersonation attack of a vehicle could lead to the attacker getting access to confidential data and proprietary software available on the portal that is meant for the impersonated vehicle. In addition, since a vehicle is susceptible to physical attacks, there is a risk that an attacker can extract, e.g., authentication and encryption keys stored in the ECUs. Using these keys, the attacker can impersonate a vehicle or eavesdrop encrypted communication. An attacker could also access private data and proprietary software stored in the vehicle or sent over the communication link.

8. **Denial of Service**
   An attacker can execute a DoS attack targeting the portal, the communication link, or the vehicle causing software updates to fail or diagnostics to report incorrect values. As a consequence, legitimate users can be prevented from updating potentially vulnerable software. Furthermore, this attack could cause damage to the vehicle or injury to a person in the vehicle.

Based on these risks, we develop a set of security requirements which is presented in the next section.

5 Guidelines for a Secure Wireless Infrastructure

For the wireless infrastructure, the portal is communicating with a vehicle over the Internet and over-the-air. This infrastructure is also divided into three regions: *portal*, *communication link* and *vehicle*, as illustrated in Fig. 2. For each of the three regions we define a set of security requirements and discuss what protection is offered if the requirements are met. Several security requirements might seem obvious for Internet traffic and high-end Internet servers but for low-performance devices and special-purpose networks those security requirements are often lacking (cf. the complete lack of security features for wireless software updates in sensor networks [16]). Therefore, our set of security requirements can be seen as guidelines for creating a secure wireless infrastructure. In the following paragraphs a brief description of each region followed by the security requirements is presented.

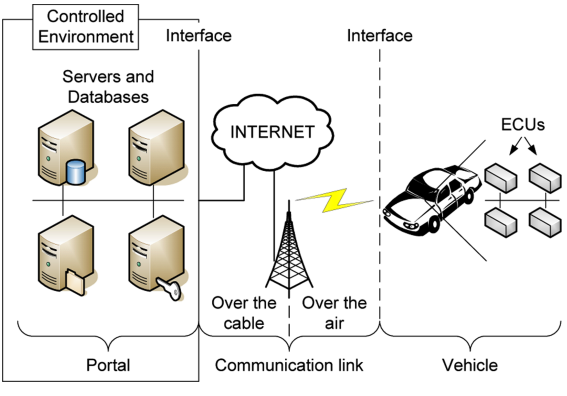

Fig. 2. Infrastructure for wireless diagnostics and software updates

5.1 Portal Security Requirements

The portal consists of servers and databases in the internal portal network and has an interface to the Internet. It has access to file servers with proprietary software that is to be installed in the vehicles and databases containing information about vehicles and what hardware and software versions they contain. Furthermore, the

portal has access to databases that contain cryptographic keys for authentication with the vehicles. Thus, the portal has access to sensitive data.

1. **Preventing Impersonation of Portal**
   Security Requirement: The portal must possess something *unique* that can be used to *establish its identity*. Certificates suitable for the vehicle environment [17] should be used, and the public key of the portal should be installed in vehicles during manufacturing. A method for handling certificate revocation must also be incorporated [18]. The portal must ensure that data sent from the portal cannot be spoofed or modified. Sensitive data should, for example, be *signed* using the portal's private key.

   Achieved Security: Prevents an outsider attacker from forging the portal identity and impersonating the portal to set up communication links with the vehicle.
2. **Intrusion Protection**
   Security Requirement: The portal is a traditional environment in the sense that it consists of powerful devices, and there already exist a number of *best practices* for firewalls, logging, and Intrusion Detection Systems (IDS) [19] that should be used.

   Achieved Security: A firewall and an IDS assist in preventing and alerting on intrusion attempts on the portal.

5.2 Communication Link Security Requirements

The communication link connects the portal to the vehicle, and is divided into two parts: *over-the-cable* and *over-the-air*. The security requirements for the over-the-cable and the over-the-air communication are the same.

3. **Secure End-to-End Communication**
   Security Requirement: A *secure end-to-end channel* for diagnostics and software updates [20] must be established. Data traffic should be encrypted and integrity protected using, for example, the TLS protocol. Cryptographic algorithms must be chosen carefully to agree with the limited resources in the vehicle. A comparison of cryptographic algorithms for use in a vehicular environment is found in [13], and a performance evaluation of public-key cryptosystem operations in WTLS is found in [21]. Moreover, the software and diagnostics requests should use *timestamps* or other methods to guarantee freshness.

   Achieved Security: Secure end-to-end communication prevents packets from being read, injected, modified and replayed by both insider and outsider attackers.

5.3 Vehicle Security Requirements

The vehicle contains sensitive data, such as cryptographic keys and proprietary software, stored in the ECUs. Furthermore, received diagnostics requests and software are executed respectively installed in the ECUs.

4. **Preventing Impersonation of Vehicle**
   Security Requirement: The vehicle must possess something *unique* to *establish its identity*. An analogy is client certificates in TLS [22]. These certificates should be installed in the vehicle during manufacturing.

   Achieved Security: Prevents both insider and outsider attackers from impersonating a vehicle.
5. **Intrusion Protection**
   Security Requirement: The vehicle is a nontraditional environment in the sense that it consists of resource-constrained embedded devices. The security requirements must be adjusted accordingly. A *firewall* should be used to block incoming traffic from non-trusted parties and to allow only trusted parties to connect to the vehicle. In addition, an *IDS* should be installed in the vehicle to detect unauthorized accesses and raise alerts on intrusion attempts on the vehicle. Proper trace and network logging should be enabled.

   Achieved Security: A firewall and an IDS assist in preventing and alerting on intrusion attempts. In addition, in the event of an intrusion, log data can be analyzed and used to reconstruct the actions after the intrusion [23]. This information can be used to prevent future intrusions.

5.4 Risks of Consequences

The risks of attacks on the portal, communication link, and vehicle could be reduced by taking the proposed security requirements into consideration. If an attack is successful, the risks of consequences could be lowered by consulting the following security requirements.

6. **Preventing Execution of Arbitrary Code**
   Security Requirement: A *filter*, e.g., a blacklist, which contains a list of disallowed commands, must be used at the portal to prevent generating diagnostics requests or software that contain certain *dangerous*[3] commands that should not be allowed to be executed remotely in the ECUs. These commands should still be available when physically connecting to the vehicle. Therefore, a solution to remove these commands from the command set is not suitable. Moreover, the software and the diagnostics requests should be *signed* by the portal, and the vehicle must *verify* that the received software and requests have not been altered. The vehicle should also use a filter to prevent dangerous commands from executing remotely in the ECUs. This is comparable to server-side security enforcements for cross-site scripting attacks [24].

   Achieved Security: An insider attacker is prevented from generating and sending diagnostics requests and software that contain dangerous commands. Since these commands are not allowed in the requests and software created at the portal, they will not be executed in the ECUs. An outsider attacker is prevented from generating and modifying the software and diagnostics requests. An attacker is also prevented from executing dangerous commands in the ECUs.

[3] Commands such as triggering the airbag or turning off the head lights. Such commands must be well-defined before deployment.

7. **Secure Storage and Communication**
   Security Requirement: The portal must *encrypt* the proprietary software binaries and the private information it stores about customers using a strong symmetric cipher, e.g., AES. Access to the data requires proper authentication and authorization. Data traffic in the internal portal network should be protected using, e.g., the transport layer security (TLS) protocol. On the other hand, the vehicle should use a *tamper-resistant storage* to store sensitive data, such as encryption and authentication keys, private information and downloaded software. For example, a trusted platform module [25] could be used. Moreover, data traffic on the in-vehicle network should be protected with respect to data authentication [26].

   Achieved Security: Prevents an outsider attacker from accessing sensitive data in the portal and the vehicle. Moreover, an attacker is prevented from injecting messages as well as altering messages in the internal portal network and the in-vehicle network.
8. **Denial-of-Service Protection**
   Security Requirement: The portal and the vehicle should use proper DoS protection, although such solutions exists for traditional environments [27], they are nonexistent for vehicles. In addition, the communication protocol must be resistant to packet loss caused by not only communication problems, such as bad reception, but also intentional attacks. Therefore, a *reliable* protocol which also can handle low bandwidth communication with long delays must be used. For example, the SCTP [28] protocol provides reliable message-stream communication.

   Achieved Security: Proper DoS protection assists in preventing availability attacks. Furthermore, a reliable protocol helps for protecting the availability of the link.

6 Security Policies

Since wired diagnostics and software updates typically are performed in closed and controlled environments with immediate supervision, security policies have been nonexistent. However, for allowing wireless diagnostics and software updates, defining a set of security policies is imperative. If several parties are involved, e.g., portal, service station, and vehicle owner, it is especially important to define who is allowed to perform what actions. We provide a few policies as examples. Policies for various involved parties must be specified.

- Only the portal is allowed to create and sign software.
- The portal and service stations are allowed to perform software updates of signed software on the vehicle.
- The vehicle must verify the authenticity of the received software to verify that it was created by the portal.
- The portal and service stations are allowed to send diagnostics requests.

Furthermore, policies for the ECUs must be well-defined.

- A time limit, e.g., 30 minutes, for updating the software on the same ECU should be used.
- Only ECUs that do not affect the maneuverability of the vehicle are allowed to be updated over-the-air.
- Prerequisites for updates include the engine being turned off for at least one hour, a velocity of zero mph, and no driver or passengers in the vehicle.
- A time limit, e.g., 1 minute, for responding to repeated diagnostics requests should be used.
- Only non-safety critical diagnostics requests are allowed to be sent to safety-critical ECUs, and only safety-critical diagnostics requests are allowed to be sent to non-safety critical ECUs.

These examples are only a few of the policies that need to be defined. A thorough analysis of all the ECUs in the vehicle to classify them into safety-critical classes [29] and defining policies for the different ECUs and classes is required to properly specify security policies. These policies define the security of the ECUs and prevent attackers from installing malicious software and vehicle owners from boosting the performance in the ECUs. Moreover, combination of policies could prevent denial-of-service attacks on the ECUs and more advanced cyber attacks [30,31] targeting the safety of the vehicle. Thus, the policies are the vanguard of security and safety on the vehicle and the portal.

7 Future Work

The most pertinent issue for the near future is to scrutinize the in-vehicle network for possible entry points and weaknesses. A risk analysis of the ECUs is to be conducted, and measures to provide the necessary security are to be evaluated.

Another possible direction is to explore the possibilities of using an IDS in the vehicle. The IDS could trigger on reads and writes to security-critical data or on abnormal activities, and thus detect attacks on the vehicle. Finally, it would be interesting to investigate the possibility to include a firewall in the vehicle to prevent unwanted external accesses as well as an internal filtering service within the in-vehicle network to block accesses to certain ECUs with respect to safety. It would be highly interesting to investigate how an IDS, firewall or filtering service can be adapted to the typical vehicular communication, which is significantly different from Internet traffic.

8 Conclusion

This paper aims to deepen the awareness of security risks involved in creating an infrastructure for wireless software updates and diagnostics in vehicles and provides guidelines for improving the security. The security risks for a wireless infrastructure are first assessed. The result is used to develop a set of guidelines

for creating a secure infrastructure. The infrastructure is subdivided into the portal, the communication link, and the vehicle, and a number of security risks in each part are identified. These risks must seriously be taken into consideration when designing the infrastructure and security must be incorporated from the very start. Consequently, we have listed a number of security requirements and discussed the importance of defining security policies.

References

1. See, W.-B.: Vehicle ECU Classification and Software Architectural Implications. Technical report, Feng Chia University, Taiwan (2006)
2. Miucic, R., Mahmud, S.M.: An In-Vehicle Distributed Technique for Remote Programming of Vehicles' Embedded Software. Technical report, Electrical and Computer Engineering Department, Wayne State University, Detroit, MI 48202 USA (2005)
3. Nilsson, D.K., Larson, U.E., Jonsson, E.: Creating a Secure Infrastructure for Wireless Diagnostics and Software Updates in Vehicles. Technical report, Chalmers University of Technology, 2008:02 (2008)
4. Mahmud, S.M., Shanker, S., Hossain, I.: Secure Software Upload in an Intelligent Vehicle via Wireless Communication Links. In: Proceedings of IEEE Intelligent Vehicles Symposium, pp. 587–592 (2005)
5. Raya, M., Papadimitratos, P., Hubaux, J.-P.: Securing Vehicular Communications. IEEE Wireless Communications 13(5), 8–15 (2006)
6. Shavit, M., Gryc, A., Miucic, R.: Firmware Update over the Air (FOTA) for Automotive Industry. Technical Report 2007-01-3523, SAE (2007)
7. Miucic, R., Mahmud, S.M.: Wireless Multicasting for Remote Software Upload in Vehicles with Realistic Vehicle Movement. Technical report, Electrical and Computer Engineering Department, Wayne State University, Detroit, MI 48202 USA (2005)
8. Parrillo, L.C.: Wireless motor vehicle diagnostic and software upgrade system. U.S. patent 5442553 (1995)
9. Lightner, B., Botrego, D., Myers, C., Lowrey, L.H.: Wireless diagnostic system and method for monitoring vehicles. U.S. patent 6636790 (2003)
10. Suman, M.J., Zeinstra, M.L.: Remote vehicle programming system. U.S. patent 5479157 (1995)
11. Chen, C.-H.: Vehicle security system having wireless function-programming capability. U.S. patent 6184779 (2001)
12. Wolf, M., Weimerskirch, A., Paar, C.: Security in Automotive Bus Systems. In: Workshop on Embedded IT-Security in Cars, Bochum, Germany (November 2004)
13. Raya, M., Hubaux, J.-P.: The Security of Vehicular Ad Hoc Networks. In: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 11–21. ACM Press, New York (2005)
14. Dolev, D., Yao, A.C.: On the Security of Public Key Protocols. IEEE Transactions on Information Theory 29(2), 198–208 (1983)
15. Howard, J.D., Longstaff, T.A.: A Common Language for Computer Security Incidents (SAND98-8667) (1998), `http://www.cert.org/research/taxonomy_988667.pdf`
16. Hui, J.: Deluge 2.0 - TinyOS Network Programming Manual (2005), `http://www.cs.berkeley.edu/~jwhui/research/deluge/deluge-manual.pdf`

17. IEEE. 1609.2. Standard for Wireless Access in Vehicular Networks (2004)
18. Raya, M., Jungels, D., Papadimitratos, P., Aad, I., Hubaux, J.-P.: Certificate Revocation in Vehicular Networks. Technical report, Laboratory for computer Communications and Applications (LCA), EPFL, Switzerland, 2006. LCA-Report-2006-006.
19. US-CERT. Current Malware Threats and Mitigation Strategies (2005), http://www.us-cert.gov/reading_room/malware-threats-mitigation.pdf
20. Nilsson, D.K., Larson, U.E.: Secure Firmware Updates over the Air in Intelligent Vehicles. In: Proceedings of the First IEEE Vehicular Networking & Applications Workshop (Vehi-Mobi), pp. 380–384 (2008)
21. Levi, A., Savas, E.: Performance Evaluation of Public-Key Cryptosystem Operations in WTLS Protocol. In: Proceedings of the Eighth IEEE International Symposium on Computers and Communications, pp. 1245–1250 (2003)
22. Network Working Group. The TLS Protocol Version 1.0 (1999)
23. Nilsson, D.K., Larson, U.E.: Conducting Forensic Investigations of Cyber Attacks on Automobile In-Vehicle Networks. In: Proceedings of the First ACM International Conference on Forensic Applications and Techniques in Telecommunications, Information and Multimedia (e-Forensics). ACM Press, New York (2008)
24. Jovanovic, N., Kruegel, C., Kirda, E.: Pixy: A static analysis tool for detecting web application vulnerabilities. In: Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P), pp. 258–263 (2006)
25. Trusted Computing Group. Trusted Platform Module Specification (2003), https://www.trustedcomputinggroup.org/specs/TPM
26. Nilsson, D.K., Larson, U.E., Jonsson, E.: Efficient In-Vehicle Delayed Data Authentication based on Compound Message Authentication Codes. In: Proceedings of the IEEE 68th Vehicular Technology Conference (VTC2008-Fall) (2008)
27. Deal, R.: Cisco Router Firewall Security. Cisco Press (2004)
28. Network Working Group. Stream Control Transmission Protocol (SCTP) Specification (2006)
29. Nilsson, D.K., Phung, P.H., Larson, U.E.: Vehicle ECU Classification Based on Safety-Security Characteristics. In: Proceedings of the 13th International Conference on Road Transport and Information Control (RTIC) (2008)
30. Hoppe, T., Dittman, J.: Sniffing/Replay Attacks on CAN Buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy. In: Proceedings of the 2nd Workshop on Embedded Systems Security (WESS), Salzburg, Austria (2007)
31. Nilsson, D.K., Larson, U.E.: Simulated Attacks on CAN Buses: Vehicle virus. In: Proceedings of the Fifth IASTED Asian Conference on Communication Systems and Networks (ASIACSN). ACTA Press (2008)