A Defense-in-Depth Approach to Securing the Wireless Vehicle Infrastructure

Article *in* Journal of Networks · September 2009

DOI: 10.4304/jnw.4.7.552-564 · Source: DOAJ

2 authors, including:

Dennis Kengo Oka
Synopsys, Japan
146 PUBLICATIONS **1,206** CITATIONS

A Defense-in-Depth Approach to Securing the Wireless Vehicle Infrastructure

Dennis K. Nilsson and Ulf E. Larson
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Email: {dennis.nilsson,ulf.larson}@chalmers.se

Abstract— **The automobile industry has grown to become an integral part of our everyday life. As vehicles evolve, the primarily mechanical solutions for vehicle control are gradually replaced by electronics and software solutions forming in-vehicle computer networks. An emerging trend is to introduce wireless technology in the vehicle domain by attaching a wireless gateway to the in-vehicle network. By allowing wireless communication, real-time information exchange between vehicles and between infrastructure and vehicles become reality. This communication allows for road condition reporting, decision making, and remote diagnostics and firmware updates over-the-air. However, allowing external parties wireless access to the in-vehicle network creates a potential entry-point for cyber attackers. In this paper, we investigate the security issues of allowing external wireless communication. We use a defense-in-depth perspective and discuss security challenges and propose solutions for each of the prevention, detection, deflection, and forensics approaches. We stress the important need for applying security using the defense-in-depth principle.**

I. INTRODUCTION

The automobile industry has over the last decades significantly affected our everyday life. Today, vehicles are used in various settings such as transportation of people and goods and contribute enormously to improve our society and lifestyle. In the last decade, electronics and software have played an increasingly important role in providing vehicle functionality. Modern vehicles contain an in-vehicle network which typically consists of 50-70 electronic control units (ECUs). These ECUs are responsible for various functionality in the vehicle, ranging from small tasks such as unlocking a door or enabling cruise control to more advanced functionality such as automatic brake systems and collision warning systems. On each ECU a specific and independent firmware is run. Improved versions of the firmware are developed as bugs are detected or new functionality is added. As the new firmware is released, the vehicle owner can update the firmware for the corresponding ECU by visiting an authorized service station. To perform the firmware update, a wired connection to the vehicle is established. The new firmware is transferred to the vehicle and flashed to the ROM of the particular ECU, overwriting the old firmware. In addition to firmware updates, diagnostics can be performed on the ECUs to detect errors or to determine the cause of malfunctions. For example, if the engine does not start, the service station employee can perform diagnostics to find the cause of the problem (e.g., a faulty fuse). Moreover, diagnostics is performed in test environments to test functionality (e.g., lock and unlock the driver door) and find errors in an early phase during firmware development. These firmware updates and diagnostics 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.

Developments in wireless technology have recently allowed for improved connectivity with vehicles. Emerging trends are vehicle-to-infrastructure and vehicle-to-vehicle communication to improve the safety and comfort of the drivers. For the automobile industry, the development of a wireless vehicle infrastructure would allow exploring new solutions to existing problems and possibilities to offer a wide range of novel services to their customers. For example, firmware updates over the air (FOTA) and remote diagnostics emerge as promising possibilities. Automobile manufacturers can consequently avoid the critical and costly need for vehicle recalls and simply install a new improved firmware version over the air. The benefits for FOTA are several as presented below [1].

- Minimal customer inconvenience - There exists no need for the customer to visit a service station.
- Mass updates - It is possible to update the firmware on a whole fleet of vehicles with little effort. There is no need to physically connect to each vehicle to update the firmware.
- Faster updates - It is possible to update the firmware as soon as it is released. It reduces the time a vehicle with faulty firmware is on the streets.

There are also several benefits for performing wireless diagnostics on vehicles.

- Time saving and less inconvenience - Data can be automatically extracted from numerous vehicles with little effort. There is no need to physically connect to each vehicle and perform the diagnostics manually.
- Reduced lead times from fault to action - Possibility to analyze errors and identify causes before a vehicle arrives at a service station.
- Increased product quality - Can collect data easily from a bigger population, resulting in easier and earlier involvement of design engineers.

doi:10.4304/jnw.4.7.552-564

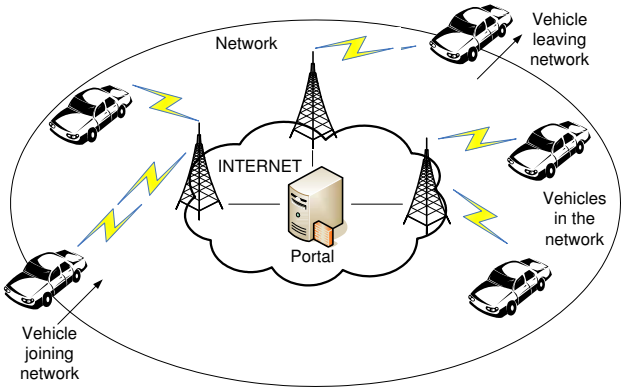

Figure 1. Wireless vehicle-to-infrastructure scenario in which a portal is communicating to a number of vehicles over a wireless link.

- Improved testing - Testers and developers can perform advanced error tracking remotely and look for error trends for a population.

However, the increased connectivity with external wireless communication leads us to believe that the wireless vehicle infrastructure in the future will be a highly plausible target for cyber attackers. Potential attacks include targeting the vehicle functionality and maneuverability which could have disastrous consequences. Since the vehicles and the surrounding infrastructure can be seen as a critical infrastructure, potentially affecting the lives of millions of drivers, we emphasize the importance of timely assessment of the security.

II. Background

In this section, we present the wireless vehicle-to-infrastructure scenario and the in-vehicle network. In addition, we describe the two main administrative functions FOTA and remote diagnostics in more detail.

A. Wireless Vehicle-to-Infrastructure Scenario

Figure 1 illustrates the wireless vehicle infrastructure scenario. Firmware updates and diagnostics requests are transmitted over a wireless link from a vehicle manufacturer's portal to a number of vehicles. The portal is part of the manufacturer's network and connected to the Internet. A wireless gateway is installed in the vehicle and connected to the in-vehicle network. The transmitted firmware and diagnostics requests are received at the vehicle by the wireless gateway and thereafter routed through the in-vehicle network to the corresponding ECU.

B. In-Vehicle Network

The in-vehicle network consists of 50-70 ECUs, gateways, and buses. A conceptual model of the in-vehicle network is illustrated in Figure 2. The ECUs are resource-constrained embedded devices with limited computational power and memory size. An ECU typically interacts with the vehicle through sensors and actuators, and communicates sensor data to other ECUs. Each ECU usually has specific responsibilities for the functionality in the vehicle. For example, one ECU is responsible for the radio system, and one ECU handles the driver door functionality (e.g., lock and window). For more complex functionality such as the engine system, a number of ECUs are cooperating. Moreover, the ECUs are connected to a shared bus, and the buses and ECUs form networks. The networks are interconnected with gateways. Common in-vehicle network types include controller area network (CAN), local interconnect network (LIN), and media oriented system transport (MOST) [2].

CAN is used for critical applications on the power train, such as the engine management system and the anti-lock braking system (ABS). A wireless gateway is connected to the CAN bus and provides access to external networks such as the Internet. LIN is a low-speed network used for non-critical tasks such as controlling automatic door locking mechanisms, power-windows, and mirrors. Other uses include smart-sensor communication for rain and darkness detection. The MOST network is a high-speed network specifically developed for multimedia and infotainment. The ECUs on a MOST network are responsible for transmitting audio, video, voice, and control data. Since the critical tasks are typically performed by ECUs on the CAN bus, we focus our attention on the security needs for the CAN bus.

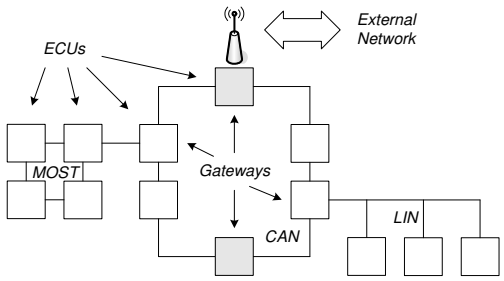

Figure 2. Conceptual model of the vehicle network consisting of CAN, MOST, and LIN networks, and a wireless gateway.

C. Administrative Functions

There exist two main administrative functions for vehicles: *diagnostics* and *firmware updates*. Diagnostics is normally used to identify defective components and faulty firmware. It affects single data parameters in the ECUs and is used for reading status (e.g., *the passenger door is locked*) or controlling activity (e.g., *unlock the passenger door*) by writing status to the ECUs [3].

The firmware update procedure involves reflashing the memory of an ECU to install new firmware. The target ECU receives the new firmware binary and flashes the binary to its ROM and reboots. After rebooting, the ECU runs the new firmware and provides new or improved functionality.

III. Need for Security

Current in-vehicle networks primarily meet reliability requirements. They are thus designed to withstand failure caused by non-malicious and inadvertent flaws which are

produced by chance or by device malfunction. Deployed protection mechanisms are therefore realized by means of fault-tolerance techniques, such as redundancy, replication, and diversity, as described in [4]. Since the in-vehicle network historically has been isolated, threats other than those against the reliability of the vehicle have not been considered. Therefore, it is fair to assume that protection against threats originating from intelligent attackers (i.e., security protection) has neither been included in the requirements nor in the design of such networks. This assumption is corroborated by a recent safety-security classification [5] which specifically emphasizes the strong relationship between safety and security and that security-related problems may well affect safety.

Exposing the in-vehicle network to external communication requires considering the following:

- The in-vehicle network and the communication channel *will* become a target for threats from malicious adversaries.
- The in-vehicle network has previously been isolated from external communication. The focus of previous research has therefore been to guarantee reliability and not security protection.
- Reliability and safety is strongly connected to security, since attacks against security may well affect safety.
- The in-vehicle network consists of resource-constrained embedded computers and the traffic patterns differ from TCP/IP-networks. Therefore traditional defense mechanisms may not work as expected, or may not work at all.

In this section, we provide our definition of security and introduce an applicable attacker model for the wireless vehicle-to-infrastructure environment. We also discuss important challenges that are specific to the in-vehicle environment which must be taken into consideration when designing security solutions.

A. Security Evaluation and Properties

Before devising security solutions, we performed a security evaluation of the wireless vehicle-to-infrastructure environment to determine the need. For the evaluation, we borrowed a set of security properties from, e.g., sensor network environments [6–9]. We believe that these properties are applicable in the vehicle setting due to the many similarities between the network types. The following five properties are considered.

- **Data Confidentiality.** The contents of messages should be kept confidential to avoid unauthorized reads.
- **Data Integrity.** Data integrity is necessary for ensuring that messages have not been modified in transit.
- **Data Availability.** Data availability is necessary to ensure that the offered service can be accessed at requested times.
- **Data Authentication.** To prevent an attacker from spoofing messages, it is important that the receiver can verify the sender of the messages.
- **Data Freshness.** Data freshness ensures that communicated messages are recent and that an attacker is not replaying old messages.

The results of the evaluation [10, 11] corroborated our assumption that the protection is insufficient. In fact, the only function present was a checksum function protecting the integrity of messages against transmission errors. Thus, the level of protection is alarmingly low.

B. Attacker Model

Based on the outcome of the security evaluation, it is reasonable to assume that a cyber attacker can perform several attacks against the communication channel and the in-vehicle network. In this section, we therefore present a definition of a cyber attacker together with a set of attack actions that it is possible to perform.

1) The Cyber Attacker Model: A cyber attacker is an individual who uses digital attacks, e.g., worms and trojan horses, to achieve a goal. This is in contrast to a physical attacker who uses physical force, e.g., using a crowbar to bend open a locked door. Furthermore, we categorize our attacker as either an *insider* or an *outsider* [12, 13]. An insider is an authorized member of a system. 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 portal or to the in-vehicle network (either through the wireless gateway or the physical On-Board Diagnostics (OBD) interface). Once access is gained, the attacker can execute attacks as an insider. As an example, consider an attacker that uses a compromised host in the portal network to obtain unauthorized access to the in-vehicle network. Once inside the in-vehicle network, the attacker sends a malicious diagnostic request to trigger the airbag, which in turn could cause injury to the driver and the vehicle to crash.

Furthermore, we assume that the attacker has sufficient knowledge to tamper with the in-vehicle network and the ECUs by the following three methods: sending diagnostics queries, sending low-level requests, and performing ECU firmware updates. Using this knowledge, the attacker can perform one or more of the following *attacker actions* [15]: *drop, flood, modify, read, spoof*, and *replay*. By reading data, an attacker can attack confidentiality (e.g., read secret keys) and privacy (e.g., read private driver information). By writing data, an attacker can attack integrity (change functionality of ECUs) and availability (disable ECUs). More detailed descriptions of the attacks are found in [10, 16, 17].

C. Security Challenges

To prevent cyber attacks on vehicles, security solutions must be designed. There exist, however, a number of fundamental limitations when designing such solutions.

- First, the ECUs inside the vehicles have limitations in computational power, memory, and bandwidth.
- Second, the ECUs operate in a real-time environment where queuing of messages and delays are typically not tolerated. The data received from sensors on a vehicle must be processed in real-time, and decisions to affect the correct actuators must be made with no imposed delay. The design of security solutions must take the real-time constraint into consideration.
- Third, the traffic patterns for vehicular communication differ from traffic patterns in traditional IP networks. For example, data on the CAN bus in the in-vehicle network is broadcast. Furthermore, automobile manufacturers could establish temporary vehicle-to-infrastructure environments for performing wireless diagnostics and firmware updates on vehicles. The different traffic patterns and communication models require different solutions. Thus, traditional solutions developed for IP networks cannot be used.

Further challenges that need consideration include verifying incoming data and protecting the wireless gateway and the in-vehicle network. Consider for example warning signals or traffic information that are exchanged between vehicles or received from intersections. It is a challenge to verify the authenticity of incoming data to a vehicle. A vehicle must assure that the received message is correct and fresh (no replay) and that it was sent from the correct vehicle or intersection.

While authenticating that incoming data is correct is one challenge, protecting the listening interface from intrusions is another. Since the wireless interface is a listening service it could possibly be subverted and allow an attacker access to the in-vehicle network. Thus, providing proper mechanisms for preventing intrusions is an important challenge. Firewalls to prevent unauthorized accesses are necessary, and logging and detection mechanisms are needed to detect and trace attackers. However, designing these security solutions to meet the real-time requirements and the limitations in the ECUs is a challenge.

A third challenge is to protect the security solutions in the in-vehicle network. Assume various cryptographic keys are used to secure the wireless communication and access control lists are used to allow only authorized connections such that the wireless gateway is protected against intrusions. An attacker could potentially access the in-vehicle network via the OBD port by physically connecting a device to the vehicle. If the security solutions protect against attacks only via the wireless gateway, an attacker could choose to attack the in-vehicle network via the OBD instead. For example, the attacker could easily extract the needed cryptographic keys and update the access control lists such that future attacks can be executed via the wireless gateway. Thus, it is a challenge to protect the in-vehicle network and the security credentials against physical attacks via the OBD.

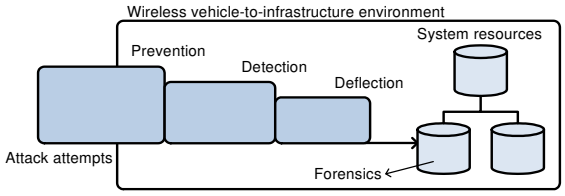

Figure 3. Defense-in-depth security approaches.

IV. Defense-in-Depth

The diversity of potential threats leads us to recommend applying a *defense-in-depth* approach to securing the wireless vehicle infrastructure. The security approaches, shown in Figure 3, are based on the defense-in-depth paradigm expressed in terms of the anti-intrusion taxonomy introduced by Halme et al. [18].

A. Prevention

The first defense approach is prevention. We first consider prevention measures for FOTA between the portal and the vehicle. The first defense approach ensures that unwanted modifications, reads, and injections of messages during the FOTA procedure are prevented. In addition, it prevents attackers from updating vehicles with malicious versions of firmware since the integrity and authenticity of firmware can be established. Moreover, we consider prevention measures for in-vehicle network communication in terms of data authentication to prevent an attacker from modifying and injecting messages in the in-vehicle network.

1) Secure Firmware-Updates-over-the-Air Protocol: We consider a secure firmware updates over the air (SFOTA) protocol that provides data authentication, data integrity, data confidentiality, and data freshness. The notation used in the protocol description is presented in Table I. We use the *encryption*, *hash*, and *signing* mechanisms without discussing the algorithms.

TABLE I.
Notation used in the protocol

PuK_P	Public key of the portal
PrK_P	Private key of the portal
K_{PA}	Symmetric encryption key shared between the portal and vehicle A
$h(x, y, z)$	One-way hash function h with inputs x, y, and z
$SIG_K(M)$	Signing of message M using the key K

a) Protocol description: Before the new firmware binary is sent to the vehicle, the binary is processed at the portal to fulfill the desired security properties. First, the new firmware binary is divided into n data fragments ($D1$ to Dn). Next, a hash chain is created by hashing each fragment in reverse order, and storing the hash with the previous fragment (Hn to $H1$). The hash calculations are

strengthened by using randomization which prevents an attacker from executing a second preimage attack [19, 20]. The values Hn to $H1$ are actually calculated by hashing a random nonce N_i chosen by the portal together with the respective data fragment and hash according to the following.

$$Hi = \begin{cases} h(N_i, Di, 0) & \text{if } i = n \\ h(N_i, Di, Hi+1) & \text{if } 1 \leq i < n \end{cases}$$

At this point, the entire hash chain is integrity-protected except for the first hash. Therefore, to provide authenticity and integrity protection of the first hash, a value denoted X containing the firmware name, version, and the number of data fragments, is concatenated with $H1$ and signed by the portal ($SIG_{PrKP}(X, H1)$). Thus, a vehicle possessing the public key of the portal can verify the authenticity and integrity of the first hash and in turn the whole hash chain. Additionally, the vehicle can verify the firmware name and version which provides freshness.

The next step is to provide confidentiality. The portal encrypts each fragment of the hash chain with an encryption key (K_{PA}), shared by the portal and the target vehicle A. This key can be generated and installed in vehicles during production, as is common practice today. The storage of the key must be considered, e.g., using secure storage. Key management for this scenario is described in [21]. Only the vehicle possessing the corresponding encryption key can decrypt the packets. We have chosen to use the encryption mode CBC (*cipher-block chaining*) which is efficient and well suited for the limited resources in the vehicle. In CBC mode, the ciphertext for a block is XORed with the next plaintext block before the block cipher encryption. As a result, each ciphertext block is dependent on all plaintext blocks processed up to that point. In addition, an initialization vector (IV) is used as input for the first block to make each message unique.

b) Procedure in the vehicle: In the vehicle the reverse procedure occurs, as illustrated by the flowchart in in Figure 4. The first ciphertext $C0$ is decrypted with the encryption key K_{PA} and the initialization vector IV. The public key PuK_P is used to verify the signature of the decrypted contents $SIG_{PrKP}(X, H1)$. The firmware name and version is compared to the current firmware, and if the version is the same or lower the packet is discarded and the procedure aborts. Otherwise, the next ciphertext $C1$ is decrypted, and the contents ($D1$ and $H2$) are hashed and verified to match the hash in the previous packet ($H1$). $H2$ is correspondingly used to verify the decrypted contents of ciphertext $C2$. The procedure continues in the same manner for the rest of the ciphertexts until all fragments have been decrypted. Due to the nature of the hash chain a vehicle can accordingly verify all the following data fragments by verifying the hash in the previous packet. The fragments are then joined to form the new firmware binary. A vehicle can verify that the received binary is unmodified and that it originated from the portal.

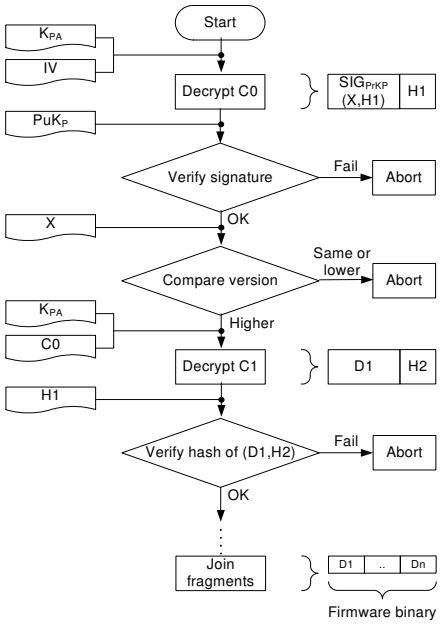

Figure 4. Flowchart describing the firmware update procedure in the vehicle.

c) Protocol security analysis: We have performed a security analysis of the proposed SFOTA protocol in terms of data authentication, data integrity, data confidentiality, and data freshness.

Data Authentication. It is assumed that the vehicle possesses the public key of the portal. Since the first packet includes a signature of the first hash, the vehicle can verify the authenticity of that hash. Furthermore, since a hash chain is created of the entire binary, the first hash also authenticates the rest of the hash chain. Thus, the authenticity and integrity of the entire binary can be verified by the vehicle.

Data Integrity. Each fragment is hashed and included with the previous fragment, forming a hash chain. Thus, the vehicle can verify that the received contents have not been modified in transit by calculating a hash of the contents and comparing it with the received hash. Due to the nature of the hash chain, all the packets in the chain are integrity-protected except for the first packet. Therefore, the first packet is signed with the private key of the portal to provide integrity protection.

Data Confidentiality. The packets are encrypted in the portal using a symmetric encryption key, and the encryption mode used is CBC. This mode of encryption is secure as long as the encryption algorithm used is secure. Thus, the entire binary is confidentiality-protected, and an attacker not knowing the encryption key cannot decrypt the packets.

Data Freshness. The firmware version included in the signed first packet guarantees freshness. Thus, extra replay protection, such as timestamps or counters, is not necessary. Since each packet contains a hash of the following packet, the vehicle can verify the order of the packets. If an attacker replays a packet later, the packet number in the replayed packet will not match

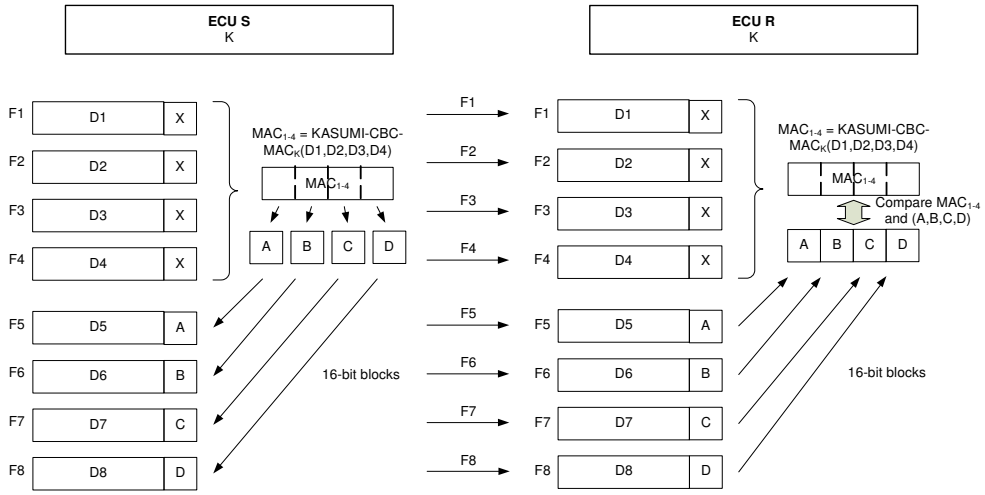

Figure 5. Compound MAC stored in the following four messages.

the current number in the chain (if we assume that the real packet has already arrived), and the replayed packet is discarded. If the packet number is modified to match the current number in the chain and replayed, the calculated hash of that packet will not match the hash in the previously received packet, and the replayed packet is thus discarded. If a packet is lost and an attacker replays that legitimate packet, the attacker would only assist in distributing the firmware. If an attacker tries to replay the entire firmware update procedure at a later time, the firmware version number in the replayed packets will be the same or lower than the currently installed firmware in the corresponding ECU, and thus the vehicle ignores those packets. We therefore argue that extra replay protection is not necessary for the firmware update procedure.

d) Wireless communication: Due to the nature of wireless communication, packets may inadvertently be lost. The *receive-verify-store* approach taken in the design of the SFOTA protocol instead of the more traditional *receive-store-verify* of the entire binary allows for a per packet verification [22]. If a packet is lost, only that packet needs to be resent as opposed to resending the entire binary, which would be very costly. Thus, the firmware update protocol is efficient and well suited for wireless communication.

2) Data Authentication using Compound Message Authentication Codes: We also consider the case where an attacker has gained access to the in-vehicle network and tries to, e.g., modify or inject messages. To prevent an attacker from performing these attacks we consider an in-vehicle network data authentication protocol.

a) Protocol description: We assume a scenario where two ECUs are communicating. For simplicity, we show the design for when ECU S is sending data frames ($F1-F8$) to the receiving ECU R. This scenario is shown in Figure 5. S and R share a MAC key K. At the sender, a MAC is calculated on a compound of four messages, divided into four parts and included in the subsequent messages. At the receiver, the MAC parts are joined and used to verify the data integrity and authenticity of the previous four messages.

To avoid making major changes to the CAN protocol, we do not modify the size of the fields. Thus, the natural approach to include security is to replace the 16-bit CRC field with a MAC field which would provide both data integrity and data authentication. However, a 16-bit MAC does not achieve a sufficient level of security [23]; therefore, we propose to compound frames four by four to achieve a total of 64 bits for the MAC.

Moreover, we assume that received messages can be queued by introducing a slight delay on the exeuction in the ECU in order to perform real-time authentication. In contrast, in [24] a slight delay is introduced in the data authentication to ensure the real-time execution.

The procedure at the sending ECU S is illustrated on the left-hand side in Figure 5. The frames are compounded four by four, which gives four 64-bit blocks of data per compound. Four blocks of data ($D1 - D4$) are used as input to the CBC-MAC calculation together with the shared key K, which outputs a 64-bit MAC (MAC_{1-4}). Then, MAC_{1-4} is divided into four 16-bit blocks (A,B,C,D) and each block is stored in the MAC field of the subsequent four frames.

The procedure for the receiving ECU R is represented on the right-hand side in Figure 5. R stores the four messages and calculates MAC_{1-4} using CBC-MAC with the key K. The following four messages are received, and from each message the 16-bit MAC is extracted. The four 16-bit blocks A, B, C, and D are combined to form the 64-bit MAC. Next, the calculated MAC_{1-4} is compared to the received MAC (A,B,C,D), and if the verification is successful, the messages $D1$ to $D4$ are authenticated and can be processed by R.

b) Discussion: The real-time data authentication imposes a slight delay on data execution in the receiving ECU. The data authentication prevents modification and injection attacks assuming that the attacker does not know the shared MAC key. Thus, it prevents an attacker from executing arbitrary commands on an ECU.

For certain real-time critical areas in the in-vehicle network it may not be possible to relax the real-time constraints to perform real-time data authentication. As an alternative, delayed data authentication could be applied which allows detection of injected or modified messages. If a MAC failure is detected, the four corresponding messages should be logged and an alert should be generated. This alert could be sent to an incident response team who would analyze the logs and identify potential attack behavior. Moreover, if the receiving ECU stores state on the values that the received commands affect, a solution to add recovery to modification and injection attacks could be possible. If a received command is found to fail the MAC verification, the effect of that command is rollbacked such that the ECU enters the previous state prior to receiving the bad command. Designing a solution for recovery requires thorough consideration, especially to avoid introducing new attacks.

More detailed descriptions about the protocols are found in [24, 25].

3) Security Benefits: By employing a prevention approach for the wireless vehicle infrastructure a number of security benefits are achieved. The major benefits include:

- Depending on the desired security properties, the protocols used should provide, e.g., data authentication, data integrity, data confidentiality, and data freshness. This ensures that received messages are unmodified, from the correct sender and arrived at the correct time. Moreover, it prevents attackers from reading the messages.
- It is the first line of defense that keeps attackers from accessing authenticated services or communicating with their targets. It detects unauthenticated accesses which could indicate that an attacker is present.

B. Detection

Next defense approach is detection. Detective measures are taken to reveal the presence of attacks and intrusions that have compromised or circumvented preventive mechanisms. For example, if the attacker is an insider, using authentication will not prevent potential attacks. We consider the aspects of deploying an intrusion detection system within the in-vehicle network. In particular we elaborate on a suitable detection principle, detector location, and attack coverage in terms of the defined attacker actions in Section III-B.

1) Detector Principle: A specification-based [26, 27] approach for attack detection is used. Specification-based systems construct a representation of correct behavior for an entity based on the current system policy together with the expected usage of the entity. Observed behavior of the entity is compared to its specification and any deviations are reported. We believe that specification-based detection is suitable for the in-vehicle network for the following reasons:

- The CAN and CANopen communication protocols are widely used, and thus, modifications or additions are assumed to be rare.
- The in-vehicle environment is dedicated to highly specific tasks with little dynamic behavior. Therefore, modification of the communication parameters section of the object directory are also assumed to be rare.
- Specification-based detection has been successfully applied for monitoring applications and network protocols. Since ECUs communicate using a network protocol, specification-based detection is suitable for the in-vehicle environment.

A specification-based detector relies on a set of security specifications that defines accepted behavior. We decided to create security specifications for the CANopen communication model [28]. For this purpose, we used two information sources: the communication protocols and the ECU object directory communication parameters. Creating security specifications for communication protocols will help us detect malformed or dropped messages, while security specifications for ECU communication parameters will help us detect illegal attempts to transmit or receive messages. We use the CANopen draft standard 3.01 to create *protocol-level security specifications* for detecting protocol violations, and *ECU-behavior security specifications* to detect illegal ECU-behavior.

a) Protocol-level security specifications: By using the CANopen application layer protocols, we can construct security specifications for the protocol. This will allow us to detect, e.g., when value in a protocol header field is outside the defined range. This type of specification is denoted *specifications for individual fields*. Furthermore, we can detect when the actual size of an firmware update data packet differs from a specified size (*specifications for dependent fields*). Moreover, we can detect, e.g., dropped packets by comparing the sequence of messages to the sequence defined by the protocol. This type is denoted *specifications for inter-object fields*.

b) ECU-behavior security specifications: ECU-behavior security specifications are created by inspecting and recording the communication parameters in the object directory of each ECU. The communication parameters section of the object directory defines the set of supported messages (and for some protocols, also the rates with which the messages are transmitted and received). We decided to create *specifications for message transmission*, which defines the set of allowed outgoing requests or response messages. This will allow us to detect if an ECU starts transmitting unexpected messages, indicating possible compromise. Correspondingly, *specifications for message reception* regards the set of allowed incoming message indications and confirmations. Finally, *specifications for message transmission and reception rates* allow us to detect, e.g., when ECUs start transmitting messages at an unexpected rate, possibly denoting attempted denial-of-service attacks.

2) Detector Location: The CAN network is a broadcast network, and the CANopen application layer does not support unique ECU identifiers for transmitters and receivers. Therefore, once a message is on the bus, it is

impossible to know where it was produced, or where it is going to be consumed. Thus, placing the detector in the network is not feasible since it can then not determine if the source of the message is allowed to transmit, or if the destination is allowed to receive. If the detector is placed in the network, the ECU-behavior security specifications can thus not be used.

Thus, by instead placing a detector on each ECU, the ECU-behavior security specifications can be enforced since it is not necessary to look for sender or receiver identities. It is already known from the object directory of the ECU what messages are valid to transmit and receive.

3) Attack Coverage: To evaluate the attack detector we use a conceptual network model illustrated in Figure 6. The conceptual network model consists of five ECUs denoted *Source, Destination, Gateway, ECU*$_1$, and *ECU*$_2$. Legitimate messages (M) travel between the networks along the bold arrow, i.e., from Source to Destination over the Gateway. ECU$_1$ and ECU$_2$ do not participate in the communication, but may receive messages if they are properly configured. The Gateway acts as a relay, i.e., it only forwards incoming messages.

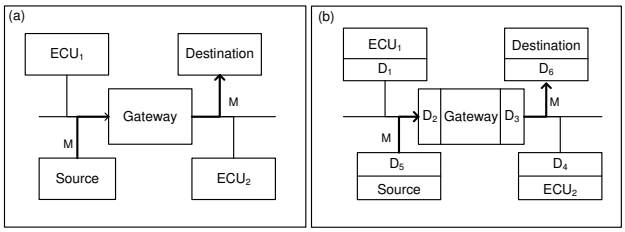

Figure 6. Conceptual network model connecting two networks through a common Gateway. Legitimate messages travel from source to destination. The left part illustrates the original network while the right part shows the network after detectors (D_i) have been added.

We then assume that the attacker can control one of ECU$_1$, ECU$_2$, or Gateway and mount the attacks discussed in Section III-B from the controlled ECU. An evaluation was performed to establish the ability of the detector to detect the mounted attacks. It was clear from the evaluation that regardless of which ECU the attacker controlled, the attacker actions were detectable by using a combination of one or more detectors D_i. Further details can be found in [11].

4) Security Benefits: Applying detection to the in-vehicle network provides a number of security benefits. The major benefits include:

- Disallowed communication patterns and ECU behavior can be detected. Without the detector, this is not possible.
- An alerting capability is added to the in-vehicle network. This is critical in notifying, e.g., the driver that there is something wrong with the vehicle. This may lead to that serious accidents are prevented, i.e., the driver can quickly pull over to the roadside and stop.

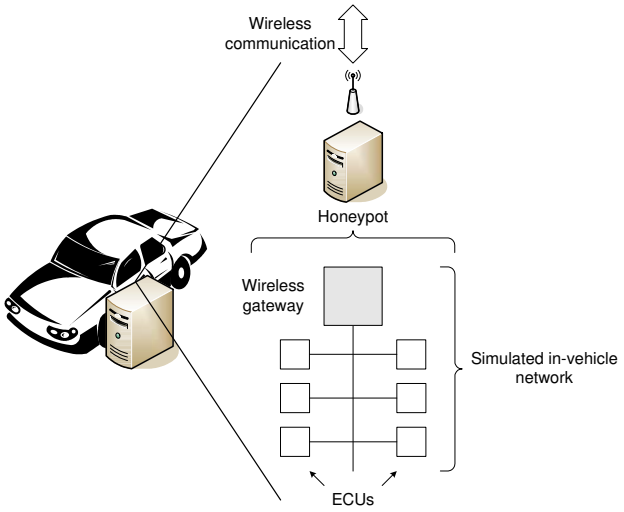

Figure 7. A single vehicle honeypot with a wireless gateway simulating the in-vehicle network.

C. Deflection

Deflection is a means of diverting attackers from the valuable assets to a faux environment where their techniques and methods can be studied. This will provide increased knowledge regarding the capabilities of the attacker and will aid construction of protective measures. A compromised vehicle constitutes a serious health concern for the driver, and therefore, it is critical that information regarding possible attacks is collected already *before* the systems are deployed.

1) Techniques for Designing a Vehicle Honeypot: The purpose of a vehicle honeypot is to simulate an in-vehicle network to attract attackers who believe that they interact with a real vehicle. Thus, the honeypot is placed in a vehicle that drives around some area, e.g., a city. During this *driving session*, the vehicle honeypot gathers the data that is sent to the wireless gateway. It should be noted that the honeypot should not communicate with the infrastructure but act as a listening device, waiting for incoming connections. After a set of driving sessions, collected data from several vehicle honeypots can then be processed and analyzed at a central location, where an analyst or operator can study attacker behavior and identify attack techniques on the in-vehicle network.

a) Vehicle honeypot design: The vehicle honeypot simulates the in-vehicle network of a single vehicle, as illustrated in Figure 7. A wireless gateway is connected to the honeypot, and thus the entry point to the honeypot is a wireless interface similar to that of the real in-vehicle network. The aim of this honeypot is to trap the behavior of attackers who gain access via the wireless gateway. We emphasize that the honeypot should be implemented in separate computer hardware placed in the vehicle and that the honeypot should have no interaction with the real in-vehicle network for safety and security reasons. Thus, if the honeypot were to be compromised, it should not allow an attacker to interfere with the real in-vehicle network.

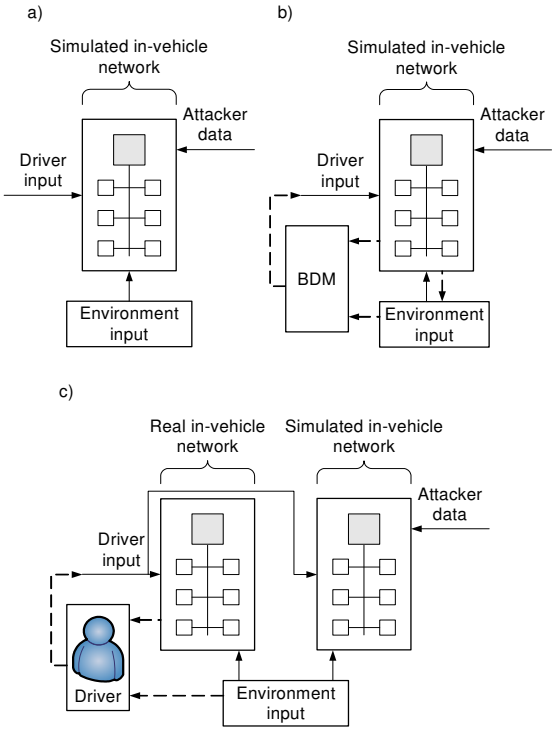

Figure 8. Vehicle simulation models.

The vehicle honeypot should appear as an authentic in-vehicle network to the attacker. Thus, the simulated ECUs should provide full functionality and behave as real ECUs. Tools, such as CANoe [29], exist for simulating ECU functionality and building simulated in-vehicle networks. We elaborate further on three different models for simulating the system to various degree of realism. These models are illustrated in Figure 8.

Each model consists of a simulated in-vehicle network, where activity is generated by a combination of driver input (e.g., accelerate, brake), environment input (e.g., sensor readings), and attack input (commands or data generated by the attacker). The first model is illustrated in Figure 8a). In this model, the driver and environment input is prerecorded and input independently of the activity within the simulated in-vehicle network and attacker input. This model is simple and has no protection against evasion or detection. The second model is illustrated in Figure 8b). In this model, the driver input is determined by a Behavior Driver Model, which accepts feedback from the simulated in-vehicle network and the environment. This may be understood as the difference between obtaining information by vehicle sensors and inputs by direct observation of the environment. The environment is simulated by a *traffic simulation model*, as discussed in [30]. Thus, this model provides a more realistic simulation in which the driver appears to react on input from the environment and the attacker. Finally, the third model is illustrated in Figure 8c). In this model, the input from the driver and the environment are generated by a real driver and the real sensors deployed in the vehicle. This creates a highly realistic model, but since we need to restrict the honeypot from the real in-vehicle network the driver cannot react to any activity generated by the attacker. This model is thus less robust against evasion and detection.

By simulating full functionality in the ECUs and replaying authentic prerecorded driver actions, the vehicle honeypot should appear as an authentic in-vehicle network to an attacker. To invite attackers, the authentication procedure or the access control on the wireless gateway could be deliberately flawed or misconfigured by the operator. Connection attempts on the wireless gateway should be considered suspicious, and data sent through the wireless gateway could potentially contain attack code.

b) Data processing and analysis: The data gathered by the vehicle honeypot is processed and analyzed at a processing center. An operator analyzes data gathered from several vehicle honeypots to detect attack trends or track attack patterns.

To extract attack commands sent on the in-vehicle network, the operator needs to separate legitimate in-vehicle communication from attack commands. Thus, the gathered data from the vehicle honeypot is processed to simplify the analysis of suspicious commands. The operator can analyze these commands and identify the type of attacks that are attempted [15]. For example, commands that involve a single request, such as *unlock the driver door*, are recognized. Moreover, the operator can identify and analyze attack code that attempt to install itself in an ECU. The attack code could also be reverse engineered to determine the effects of the attack.

The desirable outcome of the analysis is information about attack behavior, attack trends, and attacker techniques. Armed with this knowledge, automobile manufacturers can improve the security in the in-vehicle network by adding appropriate security features. A more detailed description of the vehicle honeypot is found in [31].

2) Security Benefits: The vehicle honeypot offers a number of security benefits. The major benefits include:

- Capability of collecting attack data *before* the real vehicles are attacked.
- A set of data providing important information for further design and implementation of protective measures. Furthermore, this data can be used during research to develop new algorithms capable of preventing or detecting the attacks.

D. Forensics

The forensics approach constitutes methods which are applied *after* an incident has occurred. It has been shown that current forensic methods are insufficient in a digital environment, e.g., such as the in-vehicle network, and thus new methods must be developed. We perform a classification of current types of crime against vehicles, taking into consideration access through the wireless gateway. Furthermore, we state a set of forensic design goals and propose a set of components that are necessary to meet the design goals.

1) Physical and Digital Crime and Evidence: Traditionally, only physical crimes against vehicles have been applicable, i.e., there has not been any digital assets to exploit. However, we believe that a reconsideration is in place regarding the introduction of digital equipment.

a) Types of crime against vehicles: We performed an analysis of the current threat picture against vehicles. In the following discussion, a crime is considered a *physical* crime, if the criminal is physically present at the crime scene. Furthermore, a *digital* crime is a crime where the criminal does not have to be physically present at the crime scene. We identify four types (I–IV) of crime based on the *action* taken by the criminal, e.g., cutting the brake line, and the *effect* of the action, e.g., the vehicle crashes due to lost braking capacity.

Type I crime involves a physical action resulting in a physical effect. This type targets the body of the vehicle, and involves physical theft of the vehicle, physical theft from the vehicle, or physical tampering, e.g, cutting the brake line. Type II crime involves both a physical and a digital action and results in a physical effect and is achieved by accessing and interacting with the OBD interface. Since physical access to the OBD is required, it is a physical crime. Furthermore, when the attacker is connected to the in-vehicle network via the OBD, further interaction is possible, e.g., injecting malicious code to electronically disable the brakes. This is a digital crime [10, 32].

Type III crime is crimes where a digital action causes a physical effect. This includes interaction with the wireless gateway. A criminal could access the in-vehicle network via this gateway and perform the same attacks as described above; however, no physical presence is required since the attack is purely digital. Finally, Type IV crime is entirely digital and involves no physical action and causes no physical effects. In this type, the criminal accesses the in-vehicle network via the wireless gateway and performs a digital attack that causes no physical effect. For example, the criminal could read private data such as vehicle location or modify data such as the value of the odometer.

b) Physical and digital evidence: A forensic investigation in a vehicle environment must combine physical and digital evidence gathering. Otherwise, all types of crime can not be revealed. Physical evidence includes human artifacts such as DNA, blood samples, fingerprints, and skin samples. It also includes documents or other inanimate objects which can reveal the identity of the criminal. Digital evidence includes digital images and video recordings, GPS data containing location and distance of driving, logs from door locks and window sensors, and data from driver recognition systems.

By combining physical and digital evidence it is possible to identify suspects for both physical and digital crime. For Type I crimes, physical evidence is necessary, but the investigation can be improved by combining physical and digital evidence. For example, a criminal steals a vehicle and drives to a location where the vehicle is used in another crime. After the vehicle is found, physical evidence (e.g., fingerprints) leads to a suspect. In addition, digital evidence such as location tracking from GPS data is used to find where the suspect drove. Type II crimes can be revealed by combining physical evidence from the OBD access with digital evidence from the in-vehicle network. For example, malicious code to disable the brakes at a certain velocity is sent over the OBD interface. The driver crashes and physical evidence (e.g., fingerprints on the OBD) is collected. Without digital evidence, the crash would most likely be classified as an accident. However, if digital evidence (e.g., logs from the OBD and the malicious code installed in the ECU) is gathered, it would be possible to determine when and what type of code was installed.

Type III crime may leave useful physical evidence in the vehicle and the surroundings and although the crime is digital, the investigation can be improved by combining both physical and digital evidence. However, since the root of the crime is digital, digital evidence must be collected to assist in locating a suspect. However, for this to work, data that can be used as digital evidence must be produced in the in-vehicle network. Finally, Type IV crimes necessitates the use of digital evidence since the crime has no physical effects. Thus, digital evidence must be produced and collected as mentioned above. Using digital evidence it is possible to trace the criminal and to determine the nature of the crime. However, if no digital evidence is collected, the digital crime would go unnoticed.

2) Design of a Forensic Infrastructure: For Type III and IV crimes, digital data must be collected from the in-vehicle network. Without this data, the digital investigation would not be possible. Thus, we define a set of forensic design goals and propose a set of components that are necessary to meet the design goals.

a) Forensic design goals: The value of a forensic investigation lays in its ability to discover facts regarding the five Ws, i.e., *who, what, where, when*, and *why*, as discussed by Volonino et al. [33]. Below we identify and describe three in-vehicle specific goals that must be met in order to properly conduct a forensic investigation:

- *A method to detect events in the vehicle must be present.* To perform a digital forensic investigation, an alert about a security violation must have been triggered to provide reason to initiate the forensic investigation.
- *Data to answer the questions who, what, where, when, and why must be produced and securely stored in the vehicle.* For example, messages could include a MAC (as described in Section IV-A.2) which provides proofs about the message content and the identify of the sender. During the forensic investigation, this data must be available in the network. Availability is affected by the security of the data, and thus, the data must therefore also be properly protected. An investigator should be able to extract the necessary information when needed.

- *Information about the current state (e.g., firmware versions) in a vehicle must be available and stored in a separate and secure location.* To detect whether the vehicle has been tampered with, an investigator must be able to compare the extracted data regarding the vehicle state to the original data after a security violation has occurred.

b) Infrastructure design: To detect an event at an early stage it is necessary to introduce a detection mechanism, e.g., an intrusion detection system (Section IV-B), to the in-vehicle network. The event detection requirements address what devices need to be present for detection and for alerting the appropriate authority, e.g., the driver, that a security violation has been detected.

A specification-based detection system [11] issues alerts when prohibited events occur. These alerts, together with the network events can aid investigation. In addition, there is a need for a storage device and a device for writing event and alert data to the storage. In Figure 9, the *Storage Device, Supporting ECU*, and *Detection System* are added to the in-vehicle network (cf. Figure 2).

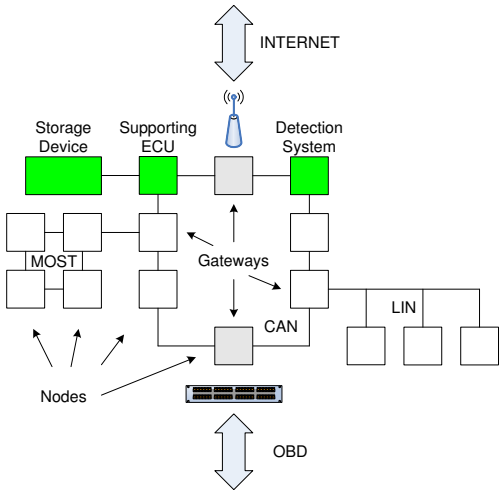

Figure 9. An in-vehicle network with a detection system and a storage device with a supporting ECU device attached.

The detection system is added to detect and alert on security violations, and the storage device together with the supporting ECU is placed in the network to contain the collected data. It is also necessary that the devices and the stored data have protection from tampering, and that reliable timestamps are produced.

c) Evaluation: To evaluate the proposed architecture, the Integrated Digital Investigation Process model [34] was applied. From the evaluation it was concluded that without a supporting forensic infrastructure, Type III and IV crimes could not be detected, but that the investigation must rather rely on physical data only. It is therefore likely that erroneous conclusions are drawn and that investigative resources are spent unnecessarily. Furthermore, detection of the crime prior to, e.g., an incident occurring relies on the fact that unexpected behavior is reported by the driver when the vehicle is in use. Further details can be found in [35, 36].

3) Security Benefits: The analyses and results produced in this section offer a number of benefits when conducting forensic investigations in in-vehicle networks. The major benefits are:

- Ability of detecting crime *before* they can cause serious harm.
- Identification of several types (I–IV) of crime of which Type III and IV cannot be readily revealed with current sources of evidence.
- A proposed infrastructure which can significantly aid the investigator during a forensic investigation of vehicle crime.

V. Discussion and Outlook

The emerging trend for ubiquitous connectivity for vehicles introduces a number of security risks. The automobile industry has traditionally dealt mainly with reliability concerns and therefore, fulfilling the future security needs in vehicles is a challenging task. A first step toward securing vehicles against cyber attacks is for the automobile industry to adopt a defense-in-depth approach. There exist challenges for securing the non-traditional vehicle-to-infrastructure communication and the in-vehicle network as resource-constrained embedded computers are involved. In particular, defensive solutions must be developed to provide sufficient security protection while respecting the resource limitations and without compromising the real-time demands of the embedded computers and the network. Thus, we have investigated the defense-in-depth approach and proposed solutions for *prevention* (a secure firmware update protocol and a compound MAC-based authentication scheme), *detection* (a specification-based IDS), *deflection* (a vehicle-based honeypot), and digital *forensics* (a forensic infrastructure).

For the next 5 years we believe that solutions for *prevention* and *detection* of attacks on the vehicle should be the main focus. It is essential to provide proper authentication mechanisms to prevent attackers from sending false data or accessing services in the vehicle. Detection is imperative for finding attacks on the vehicle, and a logging utility and an intrusion detection system should be incorporated in the wireless gateway. The many similarities between detection and forensics also promotes developing in-vehicle *forensic* techniques. Furthermore, vehicle honeypots should be developed and used for gathering attacker information. This will aid the process of developing the preventive and detective solutions.

For the next 10 years we see a need for *prevention* and *detection* of attacks in the in-vehicle network. Moreover, solutions for *deflection* and *counter-measures* should be considered. Data authentication in the communication between ECUs in the in-vehicle network is necessary to prevent attackers from injecting and modifying data. In addition, a lightweight and unobtrusive mechanism for logging and detecting is required to detect attacks in the real-time constrained in-vehicle network. Furthermore, unauthorized access attempts and intrusion attempts to the ECUs must be detected and logged by a dedicated

detection and logging process in the in-vehicle network. Furthermore, use of deflection systems must be further explored to continuously gather attacker data. Moreover, to counter active attacks in the in-vehicle network, proper countermeasure protection mechanisms must be researched. Prevention and detection are mainly passive protection mechanisms, and to prevent active attacks in the in-vehicle network an active intrusion prevention system is necessary.

For the next 20 years we believe that solid solutions for security have been developed and deployed in the vehicles. However, cyber attack attempts on vehicles will most likely continue and become more advanced as attackers learn more about the protection mechanisms and evolve. Therefore, the research focus should be on *recovery* of such attacks. To determine the cause of the attack and trace the effects of the attack, adequate data in the in-vehicle network must be generated and stored. The cyber attacks could have disastrous consequences and should be considered as severe crimes. To aid law enforcement agencies in the investigation of these crimes, solutions for performing digital forensic investigations on the in-vehicle network must be developed. Necessary data to trace and track cyber criminals must be readily available in the network. Moreover, procedures for performing forensic investigations of physical and digital crimes involving vehicles must be developed.

VI. Conclusions

In this paper, we have introduced and discussed security needs for wireless vehicle-to-infrastructure communication. We have adapted a well-known taxonomy to the vehicle setting and discussed for each of four defense-in-depth approaches the specific applicability and considerations of each approach. Furthermore, we have developed and evaluated protective techniques for each approach. The main challenge ahead is the creation of lightweight defense mechanisms that take the resource constraints of the ECUs and the specific communication patterns into consideration. The wireless vehicle environment is a critical infrastructure and security attacks may affect the safety of the driver and passengers. Therefore, a strong protection scheme taking several defense approaches into consideration is critical. We stress the importance of timely research and deployment of defensive mechanisms in all approaches of defense.

References

[1] R. Miucic and S. M. Mahmud, "An In-Vehicle Distributed Technique for Remote Programming of Vehicles' Embedded Software," Electrical and Computer Engineering Department, Wayne State University, Detroit, MI 48202 USA, Tech. Rep., 2005.

[2] Vector Informatik, "Serial Bus Systems in the Automobile: Part 1," http://www.vector-scandinavia.com/portal/medien/cmc/press/PTR/SerialBusSystems_Part1_ElektronikAutomotive_200611_PressArticle_EN.pdf, 2007.

[3] ——, "Vehicle Diagnostics: The whole story," http://www.vector-scandinavia.com/portal/medien/cmc/press/PDG/Diagnostics_Congress_ElektronikAutomotive_200703_PressArticle_EN.pdf, 2007.

[4] N. Storey, *Safety-Critical Computer Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, July 1996.

[5] D. K. Nilsson, P. H. Phung, and U. E. Larson, "Vehicle ECU Classification Based on Safety-Security Characteristics," in *Proceedings of the 13th International Conference on Road Transport and Information Control (RTIC)*, 2008.

[6] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: A secure sensor network communication architecture," in *IPSN '07: Proceedings of the 6th International Conference on Information Processing in Sensor Networks*. New York, NY, USA: ACM Press, 2007, pp. 479–488.

[7] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," in *Mobile Computing and Networking*, 2001, pp. 189–199.

[8] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A link layer security architecture for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, November 2004, pp. 162–175.

[9] D. K. Nilsson, T. Roosta, U. Lindqvist, and A. Valdes, "Key Management and Secure Software Updates in Wireless Process Control Environments," in *Proceedings of the First ACM Conference on Wireless Network Security (WiSec)*, 2008, pp. 100–108.

[10] D. K. Nilsson and U. E. Larson, "Simulated Attacks on CAN Buses: Vehicle virus," in *Proceedings of the Fifth IASTED Asian Conference on Communication Systems and Networks (ASIACSN)*. IASTED, 2008.

[11] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An Approach to Specification-Based Attack Detection for In-Vehicle Networks," in *Proceedings of the 12th IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2008.

[12] M. Wolf, A. Weimerskirch, and C. Paar, "Security in Automotive Bus Systems," in *Workshop on Embedded IT-Security in Cars*, Bochum, Germany, November 2004.

[13] M. Raya and J.-P. Hubaux, "The Security of Vehicular Ad Hoc Networks," in *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*. ACM Press, 2005, pp. 11–21.

[14] D. Dolev and A. C. Yao, "On the Security of Public Key Protocols," in *IEEE 22nd Annual Symposium on Foundations of Computer Science*, Stanford, CA, USA, 1981.

[15] J. D. Howard and T. A. Longstaff, "A Common Language for Computer Security Incidents (SAND98-8667)," http://www.cert.org/research/taxonomy_988667.pdf, 1998.

[16] D. K. Nilsson, *Securing the Wireless Vehicle-to-Infrastructure Environment: Diagnostics and Firmware Updates*. VDM Verlag Dr. Müller, November 27, 2008.

[17] D. K. Nilsson, U. E. Larson, F. Picasso, and E. Jonsson, "A First Simulation of Attacks in the Automotive Network Communications Protocol FlexRay," in *Proceedings of the First International Workshop on Computational Intelligence in Security for Information Systems (CISIS)*. Springer, 2008, pp. 84–91.

[18] L. R. Halme and K. R. Bauer, "AINT Misbehaving: A taxonomy of anti-intrusion techniques," in *Proceedings of the 18th National Information Systems Security Conferance*, October 1995, pp. 163–172.

[19] S. Halevi and H. Krawczyk, "Strengthening Digital Signatures via Randomized Hashing," in *Proceedings of the 26th Annual International Cryptology Conference (CRYPTO)*, August 2006.

[20] National Institute of Standards and Technology, "Randomized Hashing Digital Signatures," NIST Special Publication 800-106 Draft, 2007.

[21] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Low-Cost Key Management for Hierarchical Wireless Vehicle Networks," in *Proceedings of the 12th IEEE Intelligent Vehicles Symposium (IV)*, 2008.

[22] P. K. Dutta, J. W. Hui, D. C. Chu, and D. E. Culler, "Securing the Deluge Network Programming System," in *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*. New York, NY, USA: ACM Press, 2006, pp. 326–333.

[23] H. Handschuh and B. Preneel, "Minding Your MAC Algorithms," *Information Security Bulletin*, vol. 9, no. 6, pp. 213–221, 2004.

[24] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Efficient In-Vehicle Delayed Data Authentication based on Compound Message Authentication Codes," in *Proceedings of the IEEE 68th Vehicular Technology Conference (VTC2008-Fall)*, 2008.

[25] D. K. Nilsson and U. E. Larson, "Secure Firmware Updates over the Air in Intelligent Vehicles," in *Proceedings of the First IEEE Vehicular Networking & Applications Workshop (Vehi-Mobi)*. IEEE, 2008, pp. 380–384.

[26] C. Ko, G. Fink, and K. Levitt, "Automated Detection of Vulnerabilities in Privileged Programs by Execution Monitoring," in *Proceedings of the 10th Annual Computer Security Applications Conference*. Orlando, FL, USA: IEEE Computer Society Press, December 5–9, 1994, pp. 134–144.

[27] C. Ko, M. Ruschitzka, and K. Levitt, "Execution Monitoring of Security-Critical Programs in Distributed Systems: A specification-based approach," in *Proceedings of the 1997 IEEE Symposium on Security and Privacy*. IEEE, May 4–7, 1997, pp. 175–187.

[28] C. in Automation (CiA), "CANopen Application Layer and Communication Profile, CiA Draft Standard 3.01," http://www.can-cia.org/downloads/ciaspecifications/, 2006, visited August, 2007.

[29] Vector Informatik, "CANoe and DENoe 6.1," http://www.vector-worldwide.com/vi_canoe_en.html, 2007, visited August, 2007.

[30] J. J. Olstam, "A Model for Simulation of Surrounding Vehicles in Driving Simulators," Linköping Institute of Technology, Tech. Rep. LiU-TEK-LIC 2005:58, 2005.

[31] V. Verendel, D. K. Nilsson, U. E. Larson, and E. Jonsson, "An Approach to using Honeypots in In-Vehicle Networks," in *Proceedings of the IEEE 68th Vehicular Technology Conference (VTC2008-Fall)*, 2008.

[32] T. Hoppe and J. Dittman, "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.

[33] L. Volonino, R. Anzaldua, and J. Godwin, *Computer Forensics, Principles and Practices*. Upper Saddle River, NJ, USA: Prentice-Hall, 2007.

[34] B. D. Carrier and E. H. Spafford, "An Event-Based Digital Forensic Investigation Framework," in *Proceedings of the 4th Digital Forensic Research Workshop (DFRWS)*, Baltimore, MD, USA, August, 11–13, 2004.

[35] D. K. Nilsson and U. E. Larson, "Combining Physical and Digital Evidence in Vehicle Environments," in *Proceedings of the Third International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE)*. IEEE Computer Society, 2008, pp. 10–14.

[36] ——, "Conducting Forensic Investigations of Cyber Attacks on Automobile In-Vehicle Networks," *International Journal on Digital Crime and Forensics*, vol. 1, no. 2, pp. 28–41, Jan–Mar 2009.

Dennis K. Nilsson (dennis.nilsson@chalmers.se) is a Ph.D. candidate in Computer Security at Chalmers University of Technology, Sweden.

Currently, he is involved in a project with Volvo Car Corporation, where he is researching authentication and integrity principles for wireless communication with vehicles. Dennis has also conducted research at SRI International, California, USA, in the areas of security in wireless sensor networks and wireless communication. Moreover, Dennis has conducted research at Waseda University, Tokyo, Japan, in the areas of security and dependability in embedded systems. His research interests include vehicular security, wireless security, embedded security, and intrusion detection.

Dr. Ulf E. Larson (ulf.larson@chalmers.se) is a member of the Computer Security group at Chalmers University of Technology, Gothenburg, Sweden. Ulf has a Ph.D. in Computer Security from Chalmers University.

Currently, he is involved in a project with the Swedish Emergency Management Agency (SEMA) where he is investigating resource efficient data collection for intrusion detection in an overall project regarding secure and reliable computer communication in society. His research interests include data collection, intrusion detection, and digital forensics for both traditional and emerging infrastructures.