Automotive Cybersecurity for In-Vehicle Communication

By Kyusuk Han, André Weimerskirch, and Kang G. Shin

Automotive cybersecurity issues have emerged as information technologies are increasingly deployed in modern vehicles, and security researchers have already demonstrated the associated threats and risks. Although many security protocols have been proposed, they have not considered the threats posed by denial-of-service (DoS) attacks and external connectivity vulnerabilities. To alleviate this problem, we've proposed a new, secure in-vehicle communication protocol, called "ID-Anonymization for CAN (IA-CAN)." This protocol can protect against DoS attacks as well as provide a secure channel between in-vehicle components and external devices for advanced connected vehicle applications.

Vehicle Connectivity and Cybersecurity Risks

Modern cars are equipped with an average of 70 electronic control units (ECUs) that provide advanced functionality in the vehicle. These ECUs are internally connected via serial buses and communicate using a de facto standard protocol called the Controller Area Network (CAN). Recent innovations in automobile communication technology include vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) short-range communication, as well as vehicle-to-Internet communication via an embedded modem or Bluetooth-paired cell phone. Connected vehicle technology also includes connectivity to external devices such as smartphones and tablet PCs. One example is Ford Motors' OpenXC that directly extracts rich data from the vehicle (OBD-II port) and transmits the data to Android devices through a vehicle interface (VI) as depicted in Figure 1.

As vehicle connectivity becomes more common, new security risks emerge. For example, RiskIQ claimed that malicious mobile apps are becoming more prevalent, and in 2013, 12.7 percent of all Google Play apps were malicious.[1] The likelihood of successful automotive attacks increases with the number of Bluetooth-enabled vehicles that use paired smartphones, which in turn can be used as attack paths. Current vehicle systems are dreadfully vulnerable against these threats due to the lack of security considerations in the architectural design.

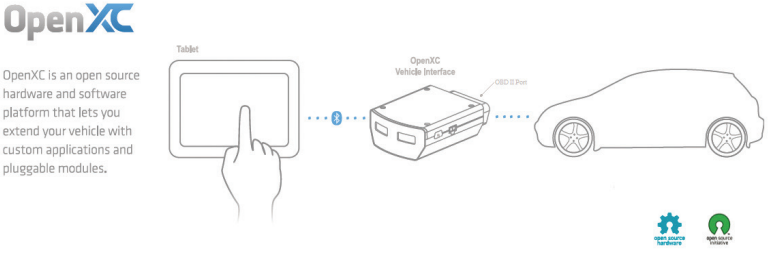

Figure 1 | Ford Motors' OpenXC (http://openxcplatform.com/).

When CAN originally became the de facto automotive standard in the 1980s, design choices were greatly influenced by strict constraints such as low cost and low network latency, while CAN security was barely considered. CAN is still used today, but the automotive landscape has drastically changed, with cars being connected through wireless interfaces and electronics being increasingly important. Security researchers have already reported the weaknesses of CAN in today's vehicles. For example, in 2010, Koscher et al. argued that CAN is insecure and vulnerable to attacks, attributing the following major drawbacks of the CAN architecture:

- There is no provision for authenticating the sender and the receiver of a frame.
- A CAN frame has no authentication field.
- The payload field in a CAN frame provides only up to 8 bytes of data.
- Current ECUs have too limited computational capability to perform a significant number of cryptographic operations.

In practice, vulnerabilities in current automotive networks are demonstrated by presenting various attack scenarios, e.g., disabling brakes, turning off headlights, and taking over steering (for cars equipped with parking assistant).[2,3] Note that other protocols, such as FlexRay, have also been introduced and deployed without addressing security.

Unfortunately, it is difficult to overhaul the entire design of this architecture to support security mechanisms due to cost. Therefore, adding security functions without compromising the current standard becomes the important industry requirement.

An attacker's behavior can be categorized into four types: ***interception, injection, modification,*** and ***interruption***.[4] Attack routes are categorized into physical access and remote access. While most practical and probable attacks are through remote access (e.g., compromising the vehicle interface in Figure 1), we digest possible attack scenarios below.[5] Table 1 shows more details.

1. **Extract keys**: After compromising an entity (the user's external device or the gateway [vehicle interface in Figure 1]), the attacker may try to extract the secret information from a user's device or from the gateway.
2. **Impersonating a user's device or a gateway**: An attacker's device may try to impersonate a user's device or a gateway.
3. **Fraudulent requests from a compromised user's device**: An attacker may compromise a user's device and then send invalid requests to the ECUs.
4. **Fraudulent requests from a compromised gateway**: An attacker may try to compromise the gateway through wired or wireless communications. He or she may then send malicious commands or codes to the gateway to read unauthorized vehicle information or to write control commands to the CAN.

State of the Art: Secure CAN

There have been several efforts to enhance the communication security in extremely constrained environments, where only up to 8 bytes are allowed for data transmission and ECUs' capabilities are limited.

- Nilsson et al. proposed to use the CRC field instead of consuming the data field in 2008. They link multiple CAN messages and use multiple 16-bit CRC fields to contain 64 bits of CBC-MAC.[6]
- Szilagyi and Koopman proposed a multicast authentication protocol by validating truncated MACs across multiple packets in 2010.[7]
- Schweppe et al. proposed a truncated MAC model that uses 4 bytes for message authentication to fit in the data field in 2011.[8]
- Groza and Murvay proposed broadcast authentication by deploying the TESLA (timed efficient stream loss-tolerant authentication) model intended for wireless sensor networks in 2012.[9]
- Hartkopp et al. proposed the flexible model that supports various conditions with time synchronization in 2012.[10]

The two glaring drawbacks of these approaches in the real-time system are: (1) receivers first accept all incoming frames irrespective of their validity; and (2) receivers need to do cryptographic computations to verify the validity of all frames, which inevitably incurs significant additional delay and becomes vulnerable to DoS attacks.

ID-Anonymization for Secure CAN

To overcome these drawbacks, we developed a concept of ID Anonymization for CAN (IA-CAN), where the frame ID is made anonymous to unauthorized entities, but identifiable by the authorized entities.[4] As shown in Figure 2, IA-CAN uses a two-step authentication process: *anonymous ID (A-ID) filtering* (step one) to check the authenticity of the sender and *message authentication* (step two) to check the validity of data. The current A-ID is generated from the previously used A-ID (initially from original ID assigned to the frame type), and shared secrets are established by using a nonce per session. The shared secrets are composed of a pre-shared key and a shared secret from a previous transmission between authorized entities.

Today, each ECU uses a CAN controller to connect to CAN. The CAN controller applies a frame (message) filter that only allows CAN frames that have one of the selected CAN IDs to pass for further processing in the ECU. The overall idea is that IA-CAN randomizes the CAN ID by using cryptographic operations. This ID is used by a receiver to select messages from the CAN bus to read. During each time period, the sender needs to reset the ID that is in use to match what the receiver is expecting; otherwise, the sent messages will be filtered out. An attacker who does not know the new ID cannot even reach an ECU, and therefore cannot mount an attack (in the same way as you cannot rob a bank if you don't know the address).

In step one, IA-CAN uses the frame filter to check the anonymous ID of each received frame. Generating A-IDs on a per-frame basis enables the authentication of the sender. Only an authorized sender or receiver can generate or identify a valid A-ID using a shared secret key and a random nonce. The receiver ECUs update their filters by pre-computing the A-ID and, upon receiving a frame, filter it. The ID is altered or anonymized on a per-frame basis and invalid frames are filtered without requiring any additional run-time computation. Since each A-ID is used only once, the attacker does not gain anything from reusing the captured A-ID (i.e., replay attacks are not possible).

Step two is designed for the potential attack scenario that a physically compromised device modifies

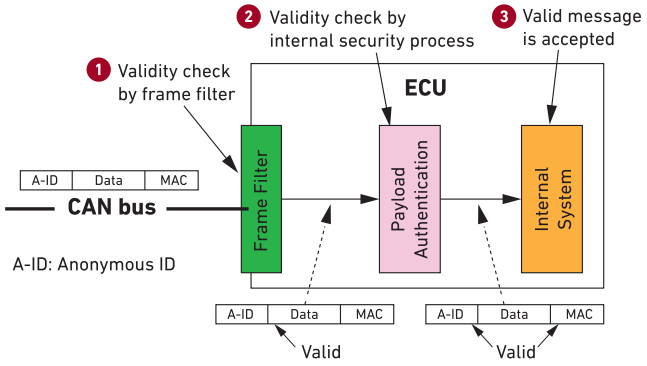

Figure 2 | Two step-authentication processes of IA-CAN.

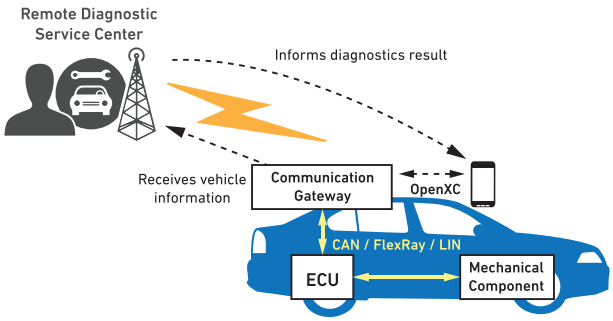

Figure 3 | An example of a recent connected vehicle application: remote diagnostic service.

messages by violating the CAN arbitration rule (a mechanism to detect and mitigate that two CAN devices send at the same time on the bus). The payload data is verified by using a cryptographic message authentication code (MAC). This prevents the attackers from modifying frames by overriding bits on the CAN. If message modification in CAN is not expected (e.g., there is only a single CAN bus that physically does not allow message modification), step two can be omitted.

The generation of the next time period's A-ID is done in idle time (while waiting for the next frame), and there is no run-time delay. The run-time overhead for step two is incurred only after the frame filter accepts the frame. While payload data authentication incurs a small run-time delay, it is still the same as the overhead of previous CAN security models.

IA-CAN ensures resiliency against DoS attacks. Two types of DoS attacks are possible for CAN: a *flooding* attack that transmits a large number of frames to a target ECU, and a *starvation* attack that disturbs transmission over the CAN bus so that ECUs can't receive frames. If a flooding attack is mounted, the latency of IA-CAN is still constant and small while the latency of existing security protocols increases linearly to the point that the ECU is blocked. Against a starvation attack, ECUs can maintain listening frames after turning into a fail-safe mode, which is a major advantage for recovery planning.

Secure Connectivity Between Vehicles and External Devices

It's increasingly common for vehicles to establish communications between external devices (e.g., a central server) and the vehicle's internal ECUs through a communication gateway, as shown in Figure 3.

We've proposed a three-step authentication protocol that provides secure communication between the external device and the ECUs in the vehicle.[5] We consider the different nature of in-vehicle network and external networks in Table 1.

As depicted in Figure 4, the protocol consists of three phases: Phase 1 (**P1**) is the initial authentication of the

Table 1 | Comparison of different entities.

Device Type	Device Lifetime	Communication	Upgrade/ Replacement Frequency	Secret Information/ Key	Key Lifetime
User device	Short-term (months – two years)	• Wireless connection over 3G/LTE, Bluetooth • e.g., smartphone, tablet, etc.	Frequent over wireless access	Users can download over wireless communication	Short-term (hours or days)
Communication Gateway	Mid-term (years)	• Connected to the user device and the CAN bus only • e.g., built-in (i.e., part of telematics) or an OBD-II dongle (as in the case of OpenXC)	Rare over limited access (mostly physical access)	Key initialization during initial purchase with system update available after physical detachment	Mid-term
INTERNAL ECU	Long-term (equal to a car's lifetime)	• Connected to the CAN bus only • e.g., internal components in the car	Only replaced when broken	Built in by manufacturer	Long-term (equal to device's lifetime)

detachable communication gateway (e.g., OBD-II dongle) over CAN. Phase 2 (**P2**) is the mutual authentication between the external entity and the gateway ECU over Bluetooth (or USB). Phase 3 (**P3**) is the authentication of the external entity's data request.

The protocol is secure against all possible attack scenarios we analyzed previously in this article. Using a short-term key as in Table 1, the risk from key extraction is limited. The gateway is considered trustworthy once it is connected to the vehicle by P1 and the user's device stores only a short-term secret. Impersonating a user's device or a gateway is prevented by P2. P3 prevents fraudulent requests from a compromised user's device and fraudulent requests from a compromised gateway.

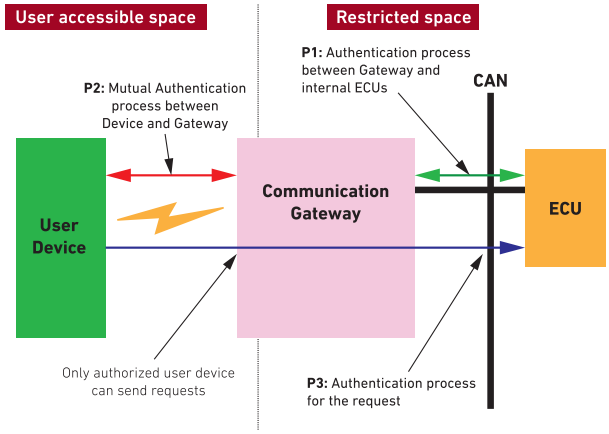

Figure 4 | Three-step authentication for secure connection between external entities (user device) and ECUs (CAN).

Conclusion

The importance of automotive cybersecurity is rapidly increasing. Although there have been efforts to implement secure solutions, many problems remain unsolved. We have introduced the IA-CAN protocol that provides strong protection against DoS attacks, and a three-step authentication protocol that provides secure integration of external devices with the vehicle's electronics. These solutions are practical automotive security approaches for in-vehicle architecture and advanced connected vehicle applications. Q

Dr. Kyusuk Han *is a Research Fellow at the University of Michigan, where his current research interest is automotive cybersecurity. He received his M.S. in Computer Engineering, Information, and Communications and his Ph.D. in Information and Communications Engineering at Korea Advanced Institute of Science and Technology (KAIST). During his Ph.D. course, Han studied security protocols for 3GPP mobile network and wireless sensor network.*

Dr. André Weimerskirch *is an Associate Research Scientist at the University of Michigan Transportation Research Institute (UMTRI). Weimerskirch holds a Ph.D. from Ruhr-University Bochum, Germany, in the area of applied data security and a Master's in Computer Science from Worcester Polytechnic Institute. Before UMTRI, Weimerskirch co-founded the automotive cybersecurity company ESCRYPT that was sold to Bosch in 2012, and was in charge of ESCRYPT's American and Asian operations with offices in the U.S., Japan, and Korea.*

Dr. Kang G. Shin *is the Kevin and Nancy O'Connor Professor of Computer Science and Founding Director of the Real-Time Computing Laboratory in the Department of Electrical Engineering and Computer Science at the University of Michigan. He received M.S. and Ph.D. degrees in Electrical Engineering from Cornell University. Shin's current research focuses on QoS-sensitive computing and networks as well as on embedded real-time and cyber-physical systems.*

REFERENCES

[1] RiskIQ. "RiskIQ Reports Malicious Mobile Apps in Google Play Have Spiked Nearly 400 Percent." Feb. 19, 2014. Retrieved from http://www.riskiq.com/company/press-releases/riskiq-reports-malicious-mobile-apps-google-play-have-spiked-nearly-400.
[2] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno. "Comprehensive Experimental Analyses of Automotive Attack Surfaces." In D. Wagner, ed. Proceedings of USENIX Security 2011. USENIX. Aug. 2011.
[3] Charlie Miller and Chris Valasek. "Adventures in Automotive Networks and Control Units." 2013.
[4] K. Han, S. D. Potluri, and K. G. Shin. "Practical Real-Time Frame Authentication for In-Vehicle Networks." escar USA 2014. June 18-19 in Ypsilanti, MI.
[5] K. Han, S. D. Potluri, and K. G. Shin. "On Authentication in a Connected Vehicle: Secure Integration of Mobile Devices with Vehicular Networks." Proceeding of ICCPS 2013, pp. 160–169. Apr. 2013.
[6] D.K. Nilsson, U.E. Larson, and E Jonsson. 2008a. "Efficient In-Vehicle Delayed Data Authentication Based on Compound Message Authentication Codes." In Vehicular Technology Conference, 2008. VTC 2008-Fall, IEEE 68th. 1–5.
[7] Chris Szilagyi and Philip Koopman. 2010. "Low Cost Multicast Authentication via Validity Voting in Time-Triggered Embedded Control Networks." In Proceedings of the 5th Workshop on Embedded Systems Security. 10. http://dx.doi.org/10.1145/1873548.1873558.
[8] Hendrik Schweppe, Yves Roudier, Benjamin Weyl, Ludovic Apvrille, and Dirk Scheuermann. 2011. "Car2X Communication: Securing the Last Meter." WIVEC 2011, 4th IEEE International Symposium on Wireless Vehicular Communications, 5-6 September 2011, San Francisco, CA, United States (June 2011), 1–5.
[9] Bogdan Groza, Stefan Murvay, Anthony van Herrewege, and Ingrid Verbauwhede. 2012. "LiBrA-CAN: a Lightweight Broadcast Authentication Protocol for Controller Area Networks." Proceedings of 11th International Conference, CANS 2012, Darmstadt, Germany. (December 2012), 185–200.
[10] Oliver Hartkopp, Cornel Reuber, and Roland Schilling. 2012. "MaCAN - Message Authenticated CAN." escar 2012, Embedded Security in Cars Conference 2012, Berlin - Germany (November 2012).