A CAN Mechanism for Security Protocol and Wireless Attack on the Car

Mrs.Ayesha Fatima
Assistant Professor,
NSAKCET,
New Malakpet, HYD.

Ms. Zubeda Begum
Assistant Professor,
NSAKCET,
New Malakpet, HYD.

Mr. Mohammmed Anwaruddin
Assistant Professor,
NSAKCET,
New Malakpet, HYD.

Dr. Mohammad Iliyas
Associate Professor,
NSAKCET,
New Malakpet, HYD.

Abstract:
Vehicle-IT convergence technology is a rapidly rising paradigm of modern vehicles, in which an electronic control unit (ECU) is used to control the vehicle electrical systems, and the controller area network (CAN), an in-vehicle network, is commonly used to construct an efficient network of ECUs. Unfortunately, security issues have not been treated properly in CAN, although CAN control messages could be life-critical. With the appearance of the connected car environment, in-vehicle networks (e.g., CAN) are now connected to external networks (e.g., 3G/4G mobile networks), enabling an adversary to perform a long-range wireless attack using CAN vulnerabilities. In this paper we show that a long-range wireless attack is physically possible using a real vehicle and malicious Smartphone application in a connected car environment. We also propose a security protocol for CAN as a countermeasure designed in accordance with current CAN specifications. We evaluate the feasibility of the proposed security protocol using CANoe software and a DSP-F28335 microcontroller. Our results show that the proposed security protocol is more efficient than existing security protocols with respect to authentication delay and communication load.

Index Terms:
Connected car, controller area network (CAN),in-vehicle network security, key management.

I. INTRODUCTION:
The newest model vehicles pursue convergence with various IT technologies to provide users with a comfortable driving environment and to effectively respond to auto emission regulations. In order to apply IT technology to vehicles, it is necessary to use a number of automotive application components. Among these components, the electronic control unit (ECU) is the most essential component that controls one or more of the electrical systems and subsystems in a vehi-cle. State-of-the art vehicular on-board architectures can consist of more than 70 ECUs that are interconnected via heterogeneous communication networks such as the controller area network (CAN), local interconnect network (LIN), or FlexRay.

As the most representative in-vehicle network, CAN has become the de facto standard because it dramatically decreases the number of communication lines required and ensures higher data transmission reliability. In this paper, we demonstrate a practical wireless attack using a real vehicle in a connected car environment, in which a driver's Smartphone is connected to the in-vehicle CAN. Our attack experiment consists of two phases: preliminary and actual attack. In the preliminary phase, i.e., before launching an actual attack, an attacker first acquires a CAN data frame to force control of the target vehicle using a diagnostic tool. In fact, the same model vehicles (more precisely, vehicles with the same configuration of automotive electronic subsystems) could be used. We note that a diagnostic tool is used to get a CAN data frame to force control of an ECU and does not need to be attached to the target vehicle during an actual attack. The attacker also manufactures a malicious self-diagnostic app that masquerades as a normal one and uploads it onto application markets. By using a self-diagnostic application such as "Torque," "Car Gauge Pro," and an OBD2 scan tool such as "EML327," "PLX KiWi," a driver can monitor CAN status information even while driving. Along with the practical wireless attack experiment, we also propose a security protocol to remedy the vulnerabilities of CAN satisfying the requirements in the following.

- The data encryption and authentication techniques ensure real-time data processing in the in-vehicle CAN.
- The method using a message authentication code (MAC) considers the limited data payload of the CAN data frame.
- Key management techniques support secure connectivity between external device and the in-vehicle CAN.

II.BACKGROUND:
A.CAN:
The CAN is a high-integrity serial data communication technology developed in the early 1980s by Robert Bosch GmbH for efficient communication between automotive applications. CAN is a multimaster broadcast communication bus system based on sender ID that allows ECUs to communicate on a single or dual wire network with data rates up to 1 Mb/s.

Fig.1. Data frame format of CAN 2.0B protocol.

The CAN protocol allowed auto makers to reduce the complexity and cost of in-vehicle network wiring. The data frame format of CAN 2.0B is shown in Fig. 1. CAN 2.0B has a 29-bit ID field divided into two parts: Base ID field and Extended ID field. The ID field is used to set the message priority. The IDE field determines the use of the 18-bit Extended ID field. The data field is a maximum of 8 bytes and includes information to be transmitted from the sender ECU to others. The cyclic redundancy check (CRC) field is used for error detection of the transferred data frame.

B.Connected Car Environment:

The connected car is receiving much attention as the next-generation Vehicle-IT convergence technology due to the rapid development of mobile communication technology and the ex-pansion of the smart device and application services. Many auto manufactures have been independently developing connected car technologies such as OnStar of GM or Connected Drive of BMW. In addition, with the popularity of a Pay-as-You-Drive insurance, a variety of electronic devices are being sold that connect to the car's OBD2 (On-Board Diagnostics) port and can be used by Smartphone applications. In general, a connected car is a vehicle that is always connected to external networks while driving. The components of a connected car are as follows:

- a vehicle with ECUs and an in-vehicle network;
- a portal to provide the vehicle with various services;
- a communication link to connect the vehicle and portal.

Fig. 2 shows a connected car environment containing these components. In a vehicle, a number of ECUs are installed and connected within CAN. The portal may be divided into Web-based and Smartphone app-based services. Recently, with the high performance and popularization of mobile communication technologies, more connected car environments are using Smartphone. Various apps supporting the connected car environment are now sold in app markets such as Google Play and the Apple App Store (e.g., Send To Car, UVO Smart Control, and BMW Connected Drive).

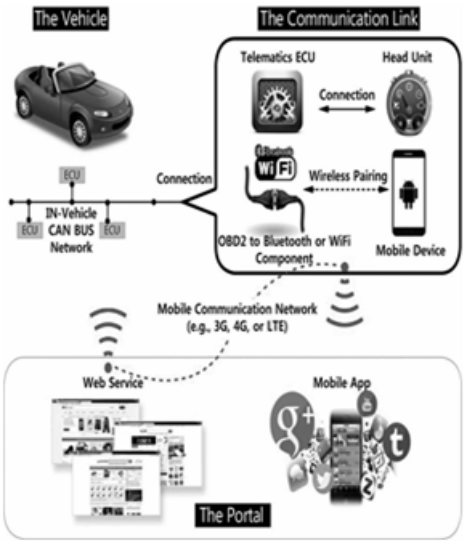

Fig.2. Connected car environment.

TABLE I: ERT CLASSIFICATION OF THE PROPOSED ATTACK MODEL

Attacker	Hacker
Tool	Automotive diagnostic tool Malicious Smartphone app
Vulnerability	Design of CAN
Action	Read Spoof
Target	Vehicle with ECUs
Unauthorized result	ECU forced control
Object	Challenge thrill

III. ATTACK MODEL AND SECURITY REQUIREMENTS:

Our attack model is the same as that of Koscheret al. in terms of exploiting the vulnerabilities of the in-vehicle CAN. However, the method of injecting malicious data into the in-vehicle CAN is clearly differentiated from those of the related work. Our attack model is designed based on an environment where a driver uses a self-diagnostic app to monitor status information after installing an OBD2 scan tool on the vehicle and then pairing it with his/her Smartphone by Bluetooth. When the driver installs on his/her Smartphone the malicious self-diagnostic app distributed by an attacker, the attacker can launch the actual attack. The attacker can obtain status information of the vehicle from the malicious self-diagnostic app and use it to inject malicious data into the in-vehicle network. Since the malicious self-diagnostic app and attacker's server communicate using the mobile communication network (e.g., 3G, 4G, or LTE), the attack is unconstrained by distance.

A. Attack Model:

According to the Computer Emergency Response Team (CERT) taxonomy suggested by, we illustrate the classification of our proposed attack model in Table I. The assumptions for our attack model are as follows.

•ATTACKER ABILITIES: An adversary has access to an automotive diagnostic tool to acquire a CAN data frame to force control of an ECU before launching an actual attack. The attacker can eavesdrop and inject the CAN data frame using a malicious self-diagnostic app into the in-vehicle CAN in the connected car environment. Thus, the attacker does not have to attack the target from a short range. The app may be widely spread through the app markets by masquerading as a legitimate self-diagnostic app for a vehicle.

•TARGET VULNERABILITIES: The target vehicle uses CAN to communicate among ECUs. CAN does not offer security services such as encryption or data frame authentication. This means that eavesdrop-ping and replay attack in CAN are possible. The unauthorized use of automotive diagnostic tools is also a security hole since the tool stores control commands for the ECUs.

•VICTIM BEHAVIOR: The victim of the target vehicle downloads the malicious self-diagnostic app to his/her smart-phone through an app market. The victim does not recognize that the app is performing malicious acts such as eavesdropping or replay attack on the in-vehicle CAN. In our proposed attack model, we do not consider an attack to compromise the ECU installed on the vehicle inside or an attack to manipulate the firmware of ECU, as these require a long period of occupancy of the target vehicle and specialized knowledge.

B. Security Requirements:

Previous work has illustrated various types of attacks possible on vehicles. We note that all these attacks eventually stem from the vulnerabilities of in-vehicle CAN, as discussed in Section II. There are three main vulnerabilities of in-vehicle CAN: 1) weak access control; 2) no encryption; and 3) no authentication. In order to construct a secure in-vehicle CAN, these three vulnerabilities have to be eliminated. However, an in-vehicle CAN is a communication environment where access control is virtually impossible. As an in-vehicle CAN is a multimaster broadcast communication environment based on the sender's ID, a connected node may receive any data frame transmitted. In addition, a malicious node may transmit a data frame by stealing the ID of a normal node (e.g., a modification and replay attack of a data frame). Since in-vehicle CAN access control is virtually impossible by the nature of the broadcast communication, it is necessary to encrypt and authenticate data frames to prevent modification and replay attack. We identify the requirements to provide a secure in-vehicle CAN as follows.

•CONFIDENTIALITY: Every data frame in CAN should be encrypted to provide confidentiality. That is, the plain text form of the data frame should be only available to a legitimate ECU or party. Due to the nature of CAN, all the data frames are broadcasted, enabling an attacker to easily eavesdrop on a CAN data frame. In particular, a data frame to force ECU control can be obtained using an automotive diagnostic tool, hence it is very easy to analyze the meaning of a relevant data frame.

•AUTHENTICATION: A control data frame in CAN is identified only by the sender ID in the data frame, which makes a replay attack possible. That is, an adversary with a valid control data frame can retransmit it, possibly masquerading as a legitimate sender. To thwart this type of attack, both the authentication and integrity of the transmitted data should be provided. The current CAN specification only offers a CRC code to ensure transmission error detection, not authentication.

TABLE II: TOOLS USED FOR THE ATTACK EXPERIMENT

Product	Model Name	Used During	Functionality and Characteristics	Product Detail Info
Diagnostic Tool	HI-DS Scanner Gold	Preliminary Phase	•Generation of ECU forced control data frames The authors of [9] used a diagnostic tool for analyzing CAN data frame	•gitauto.com Commonly used tool in auto repair shop
CAN BUS Monitoring Tool	PCAN Explorer	Preliminary Phase	•CAN data frame monitoring and capture The authors of [9] manufactured CARSHARK tool for CAN bus monitoring	•peak-system.com Commonly used tool in CAN bus system
Vehicle	Midsize car	Preliminary Phase	•Analysis and acquisition of In-Vehicle CAN data •Response analysis to ECU control data	Omitted
		Actual Attack Phase	•Used as target vehicle The authors of [9] and [19] conducted a hacking experiment using the real vehicle	
Wireless OBD2 Scan Tool	CANlink Bluetooth	Actual Attack Phase	•Support of communication between In-Vehicle CAN BUS and the app installed on the driver's smartphone	•rmcan.com Wireless CAN interface
Smartphone	Samsung Galaxy S2	Actual Attack Phase	•Install of malicious self-diagnostic app The authors of [19] introduced a short-range wireless attack model using Smartphone	•samsung.com/us/ Based on Android OS
Attacker's Server	Desktop PC	Actual Attack Phase	•Communication with malicious self-diagnostic app	Intel i5-2500 Based on Win XP

TABLE III: AUTO MANUFACTURER AUTOMOTIVE DIAGNOSTIC TOOLS

Company	Tool Name	Company	Tool Name
Audi/Volkswagen	VAS5052	Chrysler	StarScan
Ford	FoCOM	Peugeot/Citroen	XS Evolution Multiplexer
Benz	Benz XP-Star Diagnosis	HONDA	HDS
BMW	BMW ICOM A+B+C	Toyota/Lexus	Intelligent TesterII
Hyundai/KIA	HI-DS Scanner Gold	Nissan	Consult

IV. PRACTICAL ATTACK EXPERIMENT:

Based on the proposed attack model, this section describes a long-range wireless attack scenario and gives the results of our attack experiment. The tools used in the experiment are listed in Table II.

A.Preliminary Phase:

- Use of an automotive diagnostic tool to acquire CAN data frames to force control of ECUs. Exploiting an automotive diagnostic tool is very simple because the tool stores control commands for the ECUs. Global auto manufacturers offer diverse kinds of automotive diagnostic tools for convenient diagnosis. Table III shows various automotive diagnostic tools. Our acquisition process using an automotive diagnostic tool is as follows.

1)An automotive diagnostic tool is connected to the OBD2 port of a vehicle, as shown in Fig. 3(a)

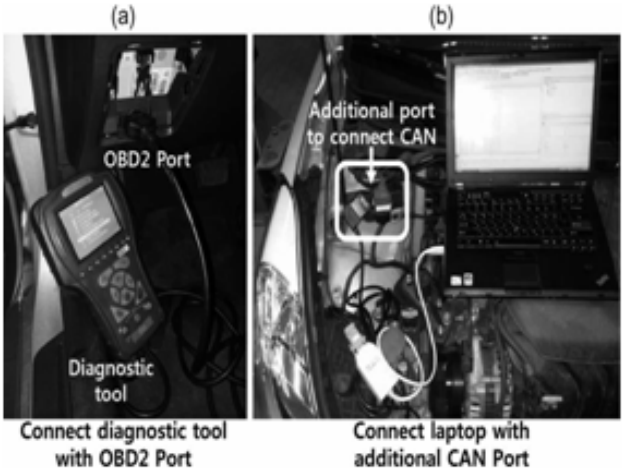

Fig.3. Experimental environment to analyze the CAN data frame.

2)The in-vehicle CAN buses are monitored after connecting a laptop to an additional port, as shown in Fig. 3(b).

3)A command is performed to forcibly actuate a certain ECU using the automotive diagnostic tool.

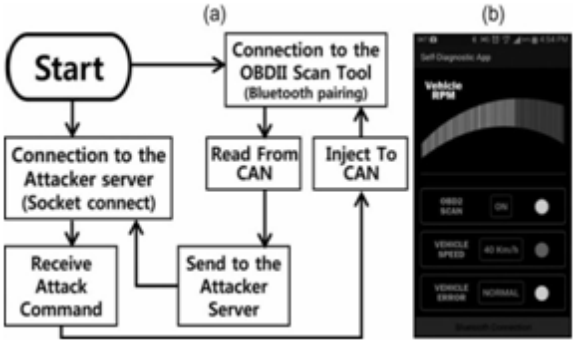

Fig.4. State diagram of the malicious self-diagnostic app and screenshot of the malicious self-diagnostic app.

Fig. 4 shows the state diagram of the malicious self-diagnostic app we produced and screenshots of the malicious self-diagnostic app.

B.Actual Attack Phase:

- ECU forced actuation attack through the malicious smart-phone app: Using an Android Smartphone, a server, an OBD2scan tool, and a midsize car, we organized an environment as shown in Fig. 5(a) and performed an experimental attack. Fig. 5(b) shows the steps of the proposed attack scenario. After the preliminary phase is completed, the actual attack phase is launched when a driver installs the malicious self-diagnostic app onto his/her Smartphone and uses it. A diagnostic tool is not physically attached to the target vehicle during the attack. The OBD2 scan tool is installed on the target vehicle and paired with the driver's Smartphone by Bluetooth. The malicious self-diagnostic app and attacker's server are then connected using a mobile communication network. The experiment was done as follows.

1)The malicious app was installed on the victim's smartphone.
2)The victim connected the Smartphone to the target vehicle using Bluetooth or Wi-Fi. The malicious app provided the victim with normal functions, masquerading as a self-diagnostic app.
3)The malicious app transmitted data frames of the in-vehicle CAN to the attacker's server using the smartphone's mobile communication network. The attacker's server checked the state of the target vehicle and trans-mitted a CAN data frame to force control of an ECU to the in-vehicle CAN via the malicious app.

4)The target vehicle had a physical malfunction caused by the abnormal control data that was transmitted from the attacker's server.

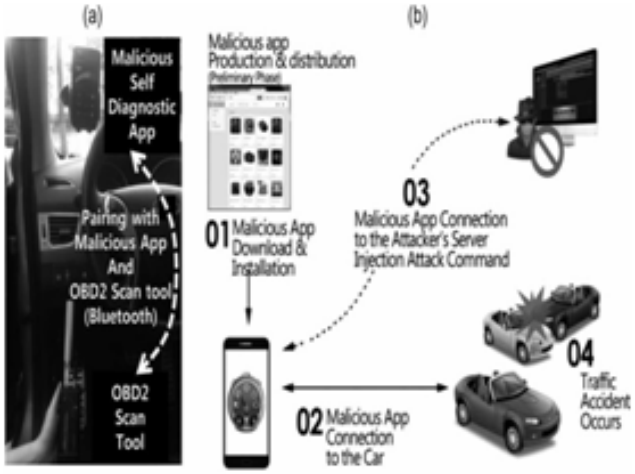

Fig. 5.Practical wireless attack experiment environment and steps of the proposed attack scenario.

V. PERFORMANCE ANALYSIS:

1) Hardware-Based Evaluation (F28335 Microcontroller):

We manufactured a Secure-ECU on a DSP-F28335 microcontroller of Texas Instruments for this evaluation. The execution time for encryption and authentication of a CAN data frame was measured by implementing the algorithms (AES-128, MAC) on the Secure-ECU firmware. Changing the CPU clock rates of the DSP-F28335 microcontroller to 150, 120, 90, and 60 MHz, we analyzed the resulting execution times of the proposed security protocol. For a more accurate evaluation, we repeated the protocol 1 000 000 times and obtained an average execution time, as shown in Fig. 11(a).

2) Software–Hardware-Based Evaluation:

To construct an evaluation environment similar to that of a real in-vehicle CAN, we used CANoe of Vector Co. CANoe is the network simula-tion software used for developing or testing embedded systems for vehicles [31]. We constructed an evaluation environment us-ing Secure-ECU, CANoe, and a VN7600 interface, as shown in Fig. 10. The proposed security protocol was also implemented on a CANoe virtual ECU node. We implemented our security protocol and built a DLL (Dynamic Linking Library) to apply it within CANoe. the newest ECUs used for vehicle development, microcontrollers with a computing power of more than 150 MHz have been installed [32] Hence, it is possible to use our proposed security protocol without data frame loss in the general in-vehicle CAN environment.

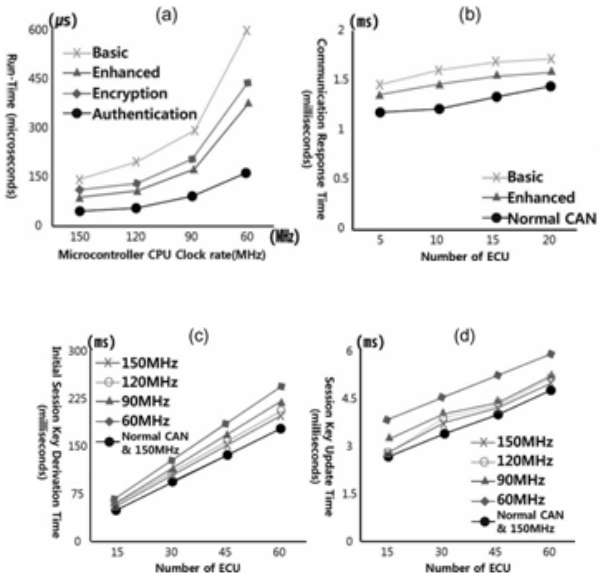

Fig.11(a) Execution time for encryption and authentication of a CAN data frame. (b) Communication response time of the in-vehicle CAN. (C) Initial session key derivation time. (d) Key update time.

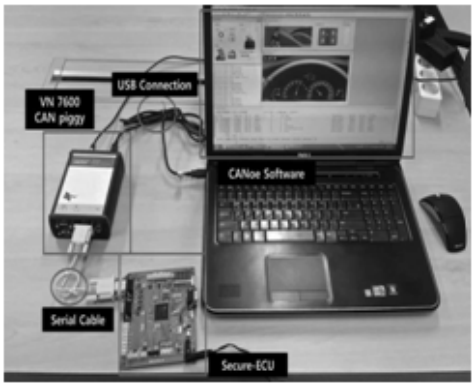

Fig. 10.Performance evaluation environment.

TABLE IV: TOOLS USED FOR THE SIMULATION

Product	Model Name	Note
Microcontroller	DSP F28335	60–150 MHz
Emulator	XDS100S	
Compiler	Code Composer	For Texas Instruments Embedded Processors
SW	CANoe	Network Simulation Tool for Vehicle
Connector	VAN 760 CAN Piggy	Interface Device

VI. CONCLUSION:

Recently, many studies on the vulnerability of in-vehicle CAN have been done. However, such attack models are un-realistic because they require significant effort and complex technology such as reverse engineering and carjacking. Thus, in this paper, we proposed an actual attack model using a malicious Smartphone app in the connected car environment and demonstrated it through practical experiments.

After demonstrating the attack model with an analysis of the vulnerability of in-vehicle CAN, we designed a security protocol that could be applied to the car environment. Furthermore, we analyzed the security and performance of the proposed security pro-tocol through an evaluation based on both Secure-ECU and CANoe. In the future, we plan to improve the performance of the proposed security protocol with an implementation of the encryption and hash algorithms on hardware to optimize our security technology.

REFERENCES:

[1]A. Saad and U. Weinmann, "Automotive software engineering and concepts," GI. Jahrestagung., vol. 34, pp. 318–319, 2003.

[2]E. Nickel, "IBM automotive software foundry," in Proc. Conf. Comput.Sci. Autom. Ind., Frankfurt, Germany, 2003.

[3]M. Wolf, A. Weimerskirch, and T. Wollinger, "State of the art: Embedding security in vehicles," EURASIP J. Embedded Syst., vol. 2007, no. 5, p. 1, 2007.

[4]R. Charette, This Car Runs on Code. [Online]. Available: http://www. spectrum.ieee.org/feb09/7649

[5]T. Nolte, H. Hansson, and L. L. Bello, "Automotive communications-past, current and future," in Proc. IEEE Int. Conf. Emerging Technol. FactoryAutom., 2005, vol. 1, pp. 992–999.

[6]K. H. Johansson, M. Torngren, and L. Nielsen, "Vehicle applications of controller area network," in Handbook of Networked and Embedded Con-trol Systems. New York, NY, USA: Springer-Verlag, 2005, pp. 741–765.

[7]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 Proc. Conf. Embedded Syst. Security, 2007, pp. 1–6.

[8]T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures," Rel. Eng. Syst. Safety, vol. 96, no. 1, pp. 11–25, Jan. 2011.

[9]K. Koscheret al., "Experimental security analysis of a modern automo-bile," in Proc. IEEE Security Privacy. Symp., Oakland, CA, USA, 2010, pp. 447–462.

[10]The EVITA project, 2008, Webpage. [Online]. Available: http:// evita-project.org

[11]H. Schweppe, Y. Roudier, B. Weyl, L. Apvrille, and D. Scheuermann, "Car2X communication: Securing the last meter—A cost-effective ap-proach for ensuring trust in Car2X applications using in-vehicle symmet-ric cryptography," in Proc. Conf. Veh. Technol., San Francisco, CA, USA, 2011, pp. 1–5.