Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 16 (2018) 47–52

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

7th International Conference on Through-life Engineering Services

Hardware Trojan Enabled Denial of Service Attack on CAN Bus

Mehmet Bozdal *[a], Maulana Randa[a,b], Mohammad Samie[a], Ian Jennions[a]

[a]*Cranfield University, College Road, Bedford, MK430AL, United Kingdom*
[b]*Indonesia Ministry of Defense, Jalan Jati No.2, Pondok Labu, Jakarta 12450, Indonesia*

Abstract

The trend of technological advances in the vehicle industry illustrates that future cars would have added functionalities with smart features, better connectivity and autonomous behaviour. These naturally involve a higher number of Electronic Control Units (ECUs) being connected using existing conventional in-vehicle network protocols such as Controller Area Network (CAN). In this context, security of systems is now becoming a major concern while industry's primary interest in the manufacturing of cars is reliability and safety. It is now in daily news that smart cars are being hacked due to weaknesses in their embedded electronics that provides ways of hardware attacks [1] [2].

Hardware Trojan (HT) is the threat that has been recently recognised as one of the primary sources of backdoor access that enables hackers to attack systems. As trouble, HT remains silent until a rare function/event triggers it for activation. This paper contributes to the challenge of demonstration of disruption in CAN buses raised from hidden Hardware Trojan. In this regard, it is presented how just a small size Hardware Trojan disrupts the CAN bus communication without an adversary having physical access to the bus. The attack is neither detectable via frame analysis, nor can be prevented via network segmentation; additionally, a rare triggering mechanism activates HT to process untraceable faults.

Keywords: controller area network, can security; electronic control unit; in-vehicle communication network; hardware trojans; reliability engineering; sensor networks

* *Corresponding author. Tel.: +44-784-6738-642.*
E-mail address: mehmet.bozdal@cranfield.ac.uk

10.1016/j.promfg.2018.10.158

1. Introduction

The automobiles are equipped with Electronic Control Units (ECUs) to control the electrical systems to improve driving comfort and safety. ECUs control most of the car's functions including safety critical engine control, airbag deployment, and anti-lock braking system. The communications between ECUs are provided by in-vehicle communication network protocols such as Controller Area Network (CAN) bus protocol. Although CAN is the most widely used in-vehicle communication network, it lacks fundamental security features like encryption and authentication. When CAN bus was initially designed in the early 1980s, security was not the main consideration because connectivity of a few devices via CAN was required and it was assumed that end users would not get access to the bus. However, modern automobiles are equipped with more than 100 ECUs [3] [4] and in some countries, On-Board Diagnostics (OBD-II) port, which connected to CAN bus, is required to be located under the dash by law for diagnostic purposes and financial reasons [5]. This; in turn, results in severe security problems.

As security of systems is becoming a significant concern, extensive researches should be directed toward vulnerabilities of the CAN. Some of these studies performed successful experimental attacks on commercial cars. The first attack on CAN protocol, attack on electric window lift, was published in 2007 by Hoppe and Dittman [6]. They later extended their work [7] and illustrated the possibility of hacking different modules including the airbag control system and gateway ECU. Kosher et al., [5] demonstrated multiple attacks, range from annoying infotainment system hack to safety critical engine stop, on commercial cars via accessing the physical network. Some researchers took advantage of the car's wireless interface for attacking systems, remotely [8]. Recently, Palanca et al. [1] implemented a selective denial-of-service (DoS) attack via plugging malicious circuit to OBD-II port. They conducted an experimental proof of concept against an unmodified 2012 Alfa Romeo Giulietta. The research shows that any person who has the physical access to CAN network can disrupt the network even with a simple tool.

Some researchers proposed solutions for preventing vulnerability of CAN bus. Lin and Sangiovanni-Vincentelli [9] proposed a software-based authentication system to prevent masquerade and replay attacks. They use ID tables, symmetric key pairs, and message counters. Although they claim it has low communication overhead, they did not consider the system maintainability in their solution hence changing a single node requires updating all nodes existing in the system. Shreejith and Fahmy [10] studied FPGA based encryption to decrease time overhead. This method can decrease the time for encryption but will increase the cost of each ECU using FPGAs. Mundhenk [3] carried out extensive research on automotive security architecture and created a probabilistic security analysis for automobile networks regarding integrity, confidentiality, and availability.

There are many CAN bus attacks reported in the literature, but none of them is an internal attack that illustrates the source of attack from inside the ECU modules connected to the CAN system. The previous attacks were achieved by either injecting malicious code on an ECU or adding a new controller or circuit to the CAN system. To our best knowledge, there is no Hardware Trojan (HT) attack implemented on CAN bus system. Although the attacks mentioned above require direct access to CAN bus or OBD-II port and/or additional circuit, HT attacks do not require the adversary to physically modify the car's system. HT is activated with a rare trigger signal; then, the affected node works as a healthy node until it gets triggered. This increases the difficulty of diagnosing the faulty node. Isolating safety critical ECUs from the user accessible network can increase the security of the vehicle, but still, it cannot remove HT threat. The solutions provided in the literature cannot handle the HT attacks due to possible integration of HT-based threats inside the CAN controller.

This paper contributes to illustrating the effects of Hardware Trojan (HT) attack on CAN bus. In this regard, we present a novel Hardware Trojan that will eliminate a CAN node from the communication system. Section 2 summarizes CAN protocol and discusses its vulnerabilities. Section 3 presents the theoretical background of the HT attack and describes the experimental setup. Section 4 discusses countermeasures and mitigating solutions followed by a conclusion.

2. Background

2.1 CAN-Bus Protocol

CAN protocol is a multi-master two-wire communication interface developed by Robert Bosh GmbH in the early 1980s and internationally standardized in ISO-11898:2003 [11]. It provides up to 1 megabit per second (bps) data transfer. CAN is a broadcast network so there is no source and destination addresses and every node can listen to any message. It has a distributed architecture which makes maintenance easier and decreases the overall system cost. Its well-recognized advantages such as high immunity to electrical interference, easy wiring, and ability to self-diagnose and repairing errors make CAN bus suitable for the automobile industry.

CAN system benefit from a two-wired differential signalling approach with a dominant logic '0' and recessive logic '1' levels. Electrically, logic '0' can overwrite the logic '1'; therefore, logic '0' is called the dominant level. There is no clock signal in CAN communication, so bit synchronization is provided via signal edges. To ensure synchronization of all nodes bit-stuffing rule is used. Bit-stuffing rule says after five consecutive bits of the same logic level, the next bit must be the complement of the previous logic level. If data has more than five successive corresponding bits, a complement bit is inserted by transmitter CAN controller. The inserted bit will be ignored by the receiver CAN controller.

SOF	11-bit or 28-bit identifier	RTR	IDE	r0	DLC	0-8 Bytes Data	CRC	ACK	EOF	IFS

Fig. 1. The bits of standard CAN bus frame

CAN protocol also has some built-in security features. Carrier Sense, Multiple Access with Collision Avoidance (CSMA/CA) rules the nodes to wait for a certain amount of inactivity before transmitting the data to sense the node is idle and collision will not occur. Collision Detection and Arbitration on Message Priority (CD+AMP) resolves the problem which multiple nodes write on the bus at the same time. The collision is resolved by message identifier bits. When a node transmits data, it listens to the bus. If message identifier bits are different from what is transmitted, a higher priority node attempts to write and takes control of the bus. There are also five error checking methods. These are; Start of Frame (SOF) single dominant bit for synchronization, Cyclic Redundancy Check (CRC) checksum of the data for data integrity, Acknowledgement (ACK) bits for successful data transmission, and End of Frame (EOF) bit for stuffing error and frame finalization. Bit-stuffing also used for error detection. If six consecutive bits of the same level occurred, an error is generated by the controller.

Another CAN security feature is Error Confinement Mechanism that utilises two error counters in each node known as Received Error Counter (REC) and Transmitted Error Counter (TEC). The value of the respective counter increases by one at the occurrence of an error during the receiving and by eight during the transmission. Then node will enter the Error Passive state if any of counter's value exceeds 127; however, its error frames will not affect the bus traffic. If TEC counter value exceeds 255, the node will be in the Buss-Off state and will no longer take part in the bus traffic. This mechanism prevents a faulty node from disrupting the whole network and controlled by the CAN controller.

2.2 CAN-Bus Vulnerabilities

Although CAN has some security features, it is vulnerable to attacks. When CAN protocol was designed, the primary purpose was connecting a few nodes with a reliable network. The bus was not readily accessible to the end user, so security was not a consideration at that time. Since that time, the automotive industry has changed dramatically. The modern cars can connect 100 nodes and the end user can easily access the CAN bus. This raises the security issue of CAN bus. Various security issues such as sniffing, replay, frame falsifying, and frame injecting attacks come from the lack of authentication and encryption. Some of the implemented attacks are shown in Table 1.

Table 1. Some of the implemented attacks on CAN bus.

Type of Attack	Interface / Environment	Attack Scenario	Reference
Replay attack	Real car / Testbench	Instrument panel lights and warning alarms are activated	[12]
Frame sniffing	Real car	Identifying gateway ECU and reaching the secure network	[7]
Frame injection	Real car / Testbench	Inject malicious code into the car's telematics	[5]
Frame falsifying	Real car	Manipulating speed data	[13]
Denial of service attack	Test bench	Sending a large volume of data to ECU and disabling the service	[14]

CAN bus is a broadcast protocol without authentication scheme. Therefore, the authenticity of the message cannot be verified. A malicious node can modify and/or insert a frame. An adversary who can access the physical bus can control almost everything that a driver can control. This may cause annoying infotainment failure to catastrophic accidents. Some researchers show that even physical access is not required and a car can be hacked remotely [8] [15]. There are also risks related to privacy issues. The modern cars are equipped with many features some of which collect data about the driver like phone contacts and location information. The lack of encryption may cause an adversary to collect the personal data and invade driver's privacy.

3. Hardware Trojan Attack on CAN Bus

3.1 Theory of Hardware Trojan Attack on CAN Bus

Systems reliability and security depends on not only the software but also the underlying hardware. Nowadays due to economic interest, embedded electronics are facing emerging hardware security threats known as Hardware Trojan (HT) [16]. HT is a stealthy modification of the hardware implemented in hardware level that remains inactive unless occurrence of a rare event. These features make HTs extremely hard to detect.

If we consider the vulnerabilities of CAN bus, HT attack can be easily implemented on CAN systems. The resulted system can be catastrophic. Imagine a CAN bus system as shown in Figure 2 where a trojanised CAN controller attached to Node 2. Because there is no authentication and broadcasting nature of CAN protocol, Node 2 can masquerade as Node 1 or any other node in the network. This will allow the malicious node to have control of the whole system. To validate this hypothesis, we implemented a DoS attack on CAN bus raised from HT.

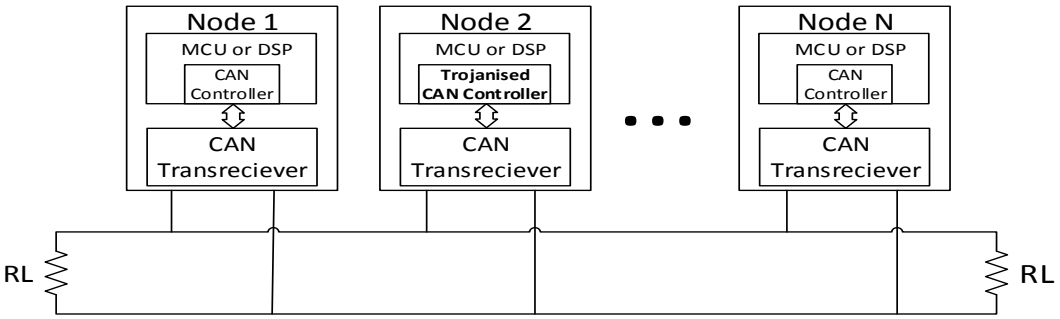

Fig. 2. Trojenised CAN controller in a CAN bus communication system

CAN is an asynchronous communication protocol, so there is no dedicated clock line. Synchronization is achieved via bit transitions. To make sure all nodes are synchronized with each other there is a limitation on the number of consecutive same level bits which is defined by stuffing rule. After five consecutive bits of the same logic level, the next bit should be the compliment. If the sixth bit is the same with previous ones, a complementary bit is inserted via CAN controller. The inserted stuff bit is ignored by the receiver CAN controller. Our Trojan will disrupt the stuffing rule and eliminate the node.

The attack is based on a physical characteristic of CAN protocol. When Trojan is triggered, the malicious node sends six logic zero bits consecutively. The transmitted logic zero is dominant; hence, it will overwrite logic one and result will be six consecutive bits at the same level. This will be against Bit-Stuffing rule and causes the error frame. Other nodes will discard the message, and error counters will be increased. When either of the error counters exceeds 256, that node will be Buss-Off state and no longer take part in the bus communication as shown in Figure 3.

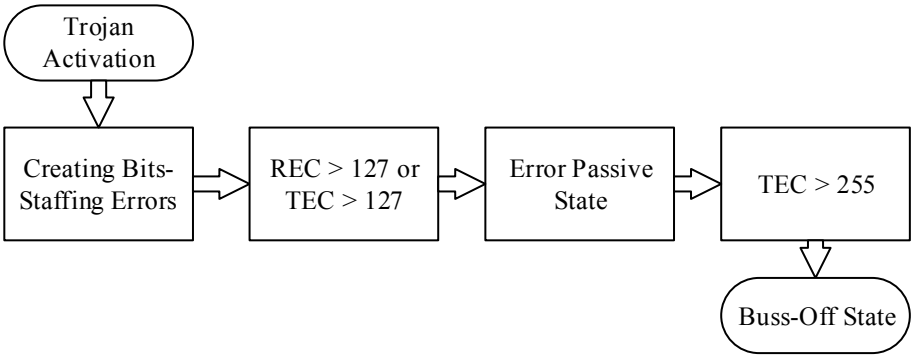

Fig. 3. The process of disabling a node

3.2 Experimental Setup

To illustrate HT in CAN system, we connect two healthy nodes and one trojenised node to each other and have a CAN network. The trojenised node waits for a trigger signal to be activated. The trigger signal is connected to the control pin of the multiplexer (Figure 4). When it is activated, it transmits manipulated data from Trojan circuit instead of original data from CAN controller.

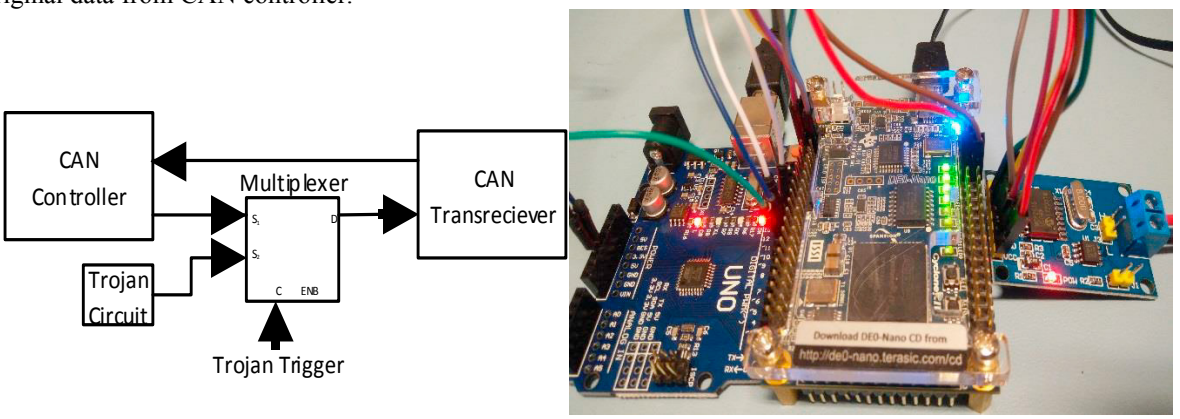

Fig. 4. (a) Hardware Trojan implementation in CAN bus controller; (b) attack implementation with Arduino UNO, Altera DE0 Nano, and Microchip MCP2551

Trojenised node is built on Altera DE0 Nano FPGA kit and connected to CAN bus via Microchip MCP2551 CAN transceiver. The other nodes are implemented on Arduino Uno microcontroller boards which are connected to Microchip MCP2551. The time-activated trigger signal is used to activate the HT. After five minutes from system start, a trigger signal is generated, and HT is activated. The Trojan circuit transmits six consecutive dominant logic '0' bits in each can frame. Other nodes reject the message and generate error flag. TEC of the trojenised node increases by eight while REC of the other nodes increases by one. When TEC exceeds 255, the node goes to Bus Off state and eliminated from the system. As a result of a successful DoS attack, requests from other nodes will not be served.

4. Counter Measures and Conclusions

Although most of the CAN attacks can be prevented by separating the network, HT is resilient to network segmentation. One possible solution can be adding a supervisory controller to the system for observing the network traffic and alert the user with any abnormalities. In this case, the supervisor controller should be adequately tested for HT. The car manufacturers should check the manufacturing process of every component they have used and apply proper tests. The end user should only use original parts verified by the manufacturers.

Attacks on CAN system has been studied and some prevention methods are proposed. However, all of these attacks are implemented via physically accessing the bus or wireless connectivity. In this paper, we implemented an internal attack scenario using HT. We eliminated a node and demonstrated a denial of service attack, but other attack scenarios can be implemented via HT which can breach the security metrics confidentiality, integrity, and availability. Therefore, authentication and encryption methods are required for a secure CAN communication system. Although encryption and authentication can solve integrity and confidentiality, availability of the network cannot be guaranteed. The future work will be focusing on how to detect HT in CAN systems and implementing a supervisory controller for anomaly detection for CAN traffic.

References

[1] A. Palanca, E. Evenchick, F. Maggi and S. Zanero, "A Stealth, Selective, Link-Layer Denial-of-Service Attack Against Automotive Networks," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2017.

[2] "A Deep Flaw in Your Car Lets Hackers Shut Down Safety Features," WIRED, 08 August 2017. [Online]. Available: https://www.wired.com/story/car-hack-shut-down-safety-features/. [Accessed 01 March 2018].

[3] P. Mundhenk, Security for Automotive Electrical/Electronic (E/E) Architectures, Göttingen: Cuvillier Verlag Göttingen, 2017.

[4] "'ECU' is a Three Letter Answer for all the Innovative Features in Your Car: Know How the Story Unfolded," embitel, 10 October 2017. [Online]. Available: https://www.embitel.com/blog/embedded-blog/automotive-control-units-development-innovations-mechanical-to-electronics. [Accessed 1 March 2018].

[5] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Snachám and S. Savage, "Experimental security analysis of a modern automobile," in *IEEE Symposium on Security and Privacy*, Berkeley, 2010.

[6] B. Groza and S. Murvay, "Security solutions for the Controller Area Network: Bringing Authentication to In-Vehicle Networks," *IEEE Vehicular Technology Magazine,* pp. 40-47, March 2018.

[7] T. Hoppe, S. Kiltz and J. Dittmann, "Security threats to automotive CAN networksPractical examples and selected short-term countermeasures," *Reliability Engineering and System Safety,* vol. 96, no. 1, pp. 11-25, 2010.

[8] S. Checkoway, D. Mccoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner and T. Kohno, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," in *Proceedings of the 20th USENIX conference on Security*, Berkeley, 2011.

[9] C.-W. Lin and A. Sangiovanni-Vincentelli, "Cyber-Security for the Controller Area Network (CAN) Communication Protocol," in *International Conference on Cyber Security*, Alexandria, VA, 2012.

[10] S. Shreejith and S. A. Fahmy, "Zero latency encryption with FPGAs for secure time-triggered automotive networks," in *International Conference on Field-Programmable Technology*, Shanghai, 2014.

[11] S. Crrigan, "Introduction to the Controller Area Network (CAN)," Texas Instrument, 2016.

[12] S. Abbott-mccune and L. A. Shay, " Techniques in hacking and simulating a modern automotive controller area network," in *IEEE International Carnahan Conference on Security Technology (ICCST)*, Orlando, 2016.

[13] R. Currie, "Hacking the CAN Bus: Basic Manipulation of a Modern Automobile Through CAN Bus Reverse Engineering," SANS Institute, 2017.

[14] S. Mukherjee, H. Shirazi, I. Ray, J. Daily and R. Gamble, "Practical DoS Attacks on Embedded Networks in Commercial Vehicles," in *Information Systems Security*, Jaipur, 2016.

[15] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," 2015.

[16] M. Tehranipoor and H. Z. X. Salmani, Integrated Circuit Authentication: Hardware Trojans and Counterfeit Detection, Storrs: Springer, 2014.