A PREFERMENT PLATFORM FOR IMPLEMENTING SECURITY MECHANISM FOR AUTOMOTIVE CAN BUS

Mabrouka Gmiden[1], Mohamed Hedi Gmiden[2] and Hafedh Gmiden[2]

[1,2]Computer and Embedded System Lab (CES),
National Engineers School of Sfax-Tunisia

ABSTRACT

The design of cryptographic mechanisms in automotive systems has been a major focus over the last ten years as the increase of cyber attacks against in-vehicle networks. The integration of these protocols into CAN bus networks is an efficient solution for leaving security level, but features of CAN bus make the performance requirements within cryptographic schemes very challenging. In the literature most of academic researches focused on designing security mechanisms for the CAN bus. Yet, very few research proposals are interested in analyzing performances requirements by using cryptographic protocols. In this paper, we investigate effects of implementing cryptographic approaches on performance by proposing an analysis methodology for implementing cryptographic approach in CAN bus communication and measuring real-time performances. Next, we propose our system which presents a tool for determining the impact of implementing of cryptographic solutions. On the other hand we have proposed an intrusion detection system using the same platform. Our tool allows the implementation of any security strategy as well as the real-time performance analysis of CAN network.

KEYWORDS

CAN bus, In-vehicle Network, Security, Analysing

1. INTRODUCTION

Nowadays, numerous in-vehicle functionalities are insured by computer components, called Electronic Control Units (ECUs) [1]. Modern cars can contain from 70 to100 of these devices [2]. At previous years, functions aboard vehicles were developed as ECUs composed of a microcontroller, sensors and actuators. With the increase number of functions such as anti-lock braking system (ABS), Electronic Stability Program (ESP), air bag, multimedia, infotainment etc. As well as with the need of these purposes to be distributed over several ECUs, communication between calculators has become a need. In order to satisfy this requirement, automakers have developed some networks like Controller Area Network (CAN), FlexRay, MOST, and LIN [3]. Today, the CAN bus has become the most widely used network in automotive applications (thanks to an excellent stability, a considerable flexibility and a low cost).

By development of automotive networks, communication between nodes has become more efficient. CAN bus is the based protocol of in-vehicle networks. But the CAN message has a broadcasted nature [4]. Moreover, the CAN protocol does not contain any authenticator field [5]. Therefore, it is easy for any attacker to full control the network message transmission, as

Dhinaharan Nagamalai et al. (Eds) : CSEIT, NCS, SPM, NeTCoM - 2018
pp. 171–186, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81813

mentioned in previous study like [6] and [7]. Until recent years, security has not been a concern in spite this clear issue.

On the other hand, vehicles have not no more been a closed machine. In fact, modern cars can connect to wired-devices like USB and CD or wireless one like 4G, smart phone and Wi-Fi, even communicate with their similar. Therefore, vehicle becomes an open system which increases the probabilities of attacks [8]. Consequently, CAN bus's security becomes a big concern and it takes over a place between recent topics for researches as well as automobile manufacturers since it threats the security of passengers as well as the safety of networks .

To address such attacks, two main layouts of security have been appeared: detection system (IDS) of attacks or anomalies on the one hand and cryptographic mechanisms to ensure confidentiality and authentication and on the other hand. Although, several researches have been oriented towards IDPS system, they have been still not 100% robust and they could not prevent all types of attacks. To exceed limitations of detective measures, many researches aim to adopt cryptographic strategies since they have been improved, in internet networks, their efficient in thwarting attacks. The challenge of designing data encryption or signature mechanisms is to protect real-time performances from being impacted.

A security mechanism is any procedure designed to prevent an attack from taking place [9]. Since the complexity of automotive systems, the implementation of a one mechanism may not frustrate all type of attacks thus the adaptation of the 'defence-in-depth' principles, which based on using of recent security mechanisms, for minimizing risks.

Our main contribution in this paper, is the design of a tool which allows, on the one hand, the implementation of a CAN bus security mechanism and the analysis, on the other hand, of real-time performances resulting from the implementation of cryptographic mechanisms. So, we deployed the same tool to develop an intrusion detection mechanism in CAN networks. The method is based on the analysis of the time intervals of the CAN message.

The remainder of this paper is organized as follows. In section II, we give a general view about automotive security issues and requirements of security solutions related to. We introduce the related work in section III. Section IV depicts the presentation of the analysis method. Section V depicts the presentation of the detection system .In Section VI, details of the proposed platform are given. We conclude in Section VII.

2. AUTOMOTIVE SECURITY ISSUES

2.1. CAN Bus Vulnerabilities

In [10], Wolf et al. show that security in CAN bus is very challenging since it cannot guarantee the following security services:

- Confidentiality: each CAN message is accessible by all nodes connected to the bus. In fact, CAN frames are transmitted in the bus according to a broadcast nature. Then, the CAN bus cannot guarantee confidentiality since an authorized node can listen to the bus and read messages.

- Authenticity: since CAN bus frame has no authentication information about the sender, an attacker connected to the bus could use the ID of any node to send a fake message.

- Availability: due to the arbitration scheme of the CAN bus, any node can put the bus in a dominant state and prevent other from sending messages which could result DoS (Denial of Service) attacks.

- Integrity: CAN protocol uses CRC (Cyclic Redundancy Check) to verify whether a message has been modified. However, this latter cannot prevent an attacker from modifying a legitimate message. In fact, she could make a correct CRC for a forged message.

- Non repudiation: in CAN protocol, it is impossible for a legitimate ECU to prove that it has sent or received a given message.

- CAN message contain between 1 and 8 bytes. So, the security protocol cannot transmit any extra authenticated data inside the classic data field (Figure 1).

- In automotive networks, the primary focus is on real-time capabilities to support control systems, which are needed to respond within a given short time. So, predictability and reliability are the dominating factors.

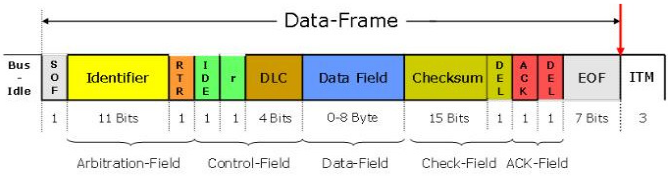

Figure 1.CAN format frame

2.2. Requirements of CAN Cus Security Solutions

- The implementation of a security mechanisms addressing CAN security issues have became urgent. However, the implementation of such solutions meets the following requirements which need to be satisfied:

- Lightweight: since in automotive system computer are very limited in computing power and memory space, heavy cryptographic functions are difficult to be performed by these ECUs. Therefore, the proposed mechanism should be as lightweight as possible.

- Respect of real-time constraints: often, applications of CAN bus are required by hard-real time constraints. Thus, security mechanism should not be impact embedded real time performances.

- Backward compatibility: the proposed mechanism should be compatible with used technologies: we are talking about retro-compatibility. On the other hand, external communications should not be prevented by the security system: we refer to t interoperability.

- Encryption: as we have seen above, a CAN data frames are easy to be eavesdropping by an attacker. So, a method of encryption should be employed in order to provide confidentiality.

- Authentication: in order to guarantee authentication of transmitted data, a hash-based message authentication code (HMAC) must be generated and transmitted along with CAN messages.

3. RELATED WORK

In [11], Nilsson et al. propose to calculate the MAC on the compounded messages then divide it into four parts and transmit them in the CRC field of the next four CAN-messages. In [12], CANauth, a lightweight authentication mechanism based on HMAC for use on the CAN bus, is proposed. The proposed authentication method is transmitted using the out-of-band CAN+ protocol, which uses 15 bytes of the CAN message as authentication data message. Both [11] and [12] give only theoretical analysis which makes it difficult to judge the performance of proposed solutions. In [13], Groza et al. propose LibraCAN, an authentication protocol based on key splitting and MAC mixing for CAN+. In Libra-CAN, the bandwidth requirements are not possible for regular CAN which makes the overhead is unacceptable. Woo et al. in [14], propose the use of AES-128 for encryption and HMAC. The proposed protocol uses 16 bits in the extended ID field and the 16-bit CRC field for transmission of 32 bits code. The implementation of the proposed protocol keeps the bus load under 50% when the CPU clock rate is 60 MHz Woo-auth provides acceptable overhead on a CAN bus. In [15], Nurnberger et al. introduce VatiCAN which enables sender and receiver ECUs to exchange authenticated data using the Keccak algorithm. Authors provide that VatiCAN guaranties a respect of real-time deadlines for safety-critical application. But it is hard to judge in case for the total system. Several security solutions proposed. However, a concrete real-time performances analysis is still limit in literature.

On the other hand, Müter et al. propose in [16] the calculation of entropy of CAN bus while the observing of traffic during a "normal" activity. If a deviation in entropy (compared to reference values) is found, an alert is then lifted. Hoppe et al. proposed IDS and demonstrated anomaly detection method by looking at frequency of messages transmitted on the bus [17].Meanwhile, authors in [18], propose an approach where each ECU has a sensor that observes the interaction of the latter with the network (sent messages but also consumed messages). Intrusion detection is based on a set of security rules based on network protocol specifications and host ECU. Intrusion detection is done in each ECU independently. Similarly, authors in [19] and [20] propose the saturation of the bus as a reaction to attacks. In [19], Miller and Valzak build a small device that plugs into the OBD-II port of a car, learns traffic patterns, and then detects anomalies. When the device detects something, it shorts circuits the CAN bus, thus disabling all CAN message. In [20], the solution presented is based on the monitoring of network traffic by each of the present ECU. When a calculator observes a message circulating on the bus, which is supposed to be its transmitter (based on the ID of the message), the ECU immediately sends an alert to crush the transmitted message. However, previous mechanisms require to be implemented in each ECU. So, they are considered as expensive solutions. Studnia et al. proposed in [21] an intrusion detection approach for an integrated automotive network. The proposed solution based on the definition of a formal language dedicated to generate a signature set for attacks aims to detect.

4. THE ANALYSIS METHODOLOGY FOR A SAFE CAN BUS COMMUNICATION

The first system, which we propose to design, aims at analyze the security performances on CAN bus network after implementing a cryptographic mechanism. In this section we want to highlight the method we used for: subsection A introduces the system model. Subsection B, explains the methodology phase and algorithms process are given in subsection C.

4.1. System Model

In this section, we introduce the system model which we adopted for implementing our method. As shown in Figure 2, our system model is composed of 2 CAN nodes connected to a CAN bus to form a network. The Node 1 with ID =0x1 send messages to Node 2 with ID =0x1.

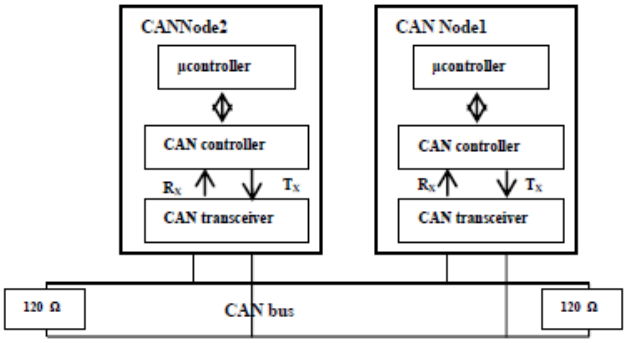

Figure 2.Synoptic diagram of system model

4.2. Fundamental Idea

The main problem on the communication side is the overhead caused by the additional data in combination with possible additional latency. Both are especially challenging when dealing with short signals requiring real time operation and low latencies. Our goal is to develop a system which can be deployed for implanting a cryptographic mechanism along with analysis real-time performances and injecting spoofed message. The proposed method allows determining the effect of security mechanism on CAN bus performances. In our work, we adapt the automotive network architecture consists of two nodes connected to CAN bus in the vehicle via a serial data communication bus. Each ECU controls a particular function of the vehicular system. The fundamental idea is to encrypt a given message in Node 1 by a cryptographic mechanism and send it to Node 2. When this latter receive the encrypted message, deploys the same mechanism to decrypt it.

4.3. Methodology Phases

Since we aim to implement cryptographic approach in the standard version of CAN protocol, the transmission process of CAN message will be different than the classic one. The whole transmission process is summarized in Figure 3. When the sender node receives a request from the receiver, it encrypts data; divides it into segments then it sends the segments via CAN bus. When the receiver gets segments, concatenates it to get complete data then it decrypts to get the original message.

4.3.1. CAN Message Encryption Phase

We need encrypt the message since we need guarantee confidentiality and integrity of automotive data network. The CAN message encryption phase is insured by encryption mechanisms and MAC methods.

4.3.2. Fragmentation Technique

As the maximum payload length allowed in the CAN data field is only 8 bytes, the available space for appending a cryptographically secure Message Authentication Code (MAC) is very limited. To solve this problem, rather than appending a MAC in one CAN frame's data field, we suggest a technique for dividing data into a size that can be stored in a message (including the sequence information) and then, each segment is transmitted.

4.3.3. CAN Message Transmission Phase

The transmission of CAN frame is carried out from the sender node to the receiver one following the CAN protocol and via CAN bus.

4.3.4. CAN message Reconstitution Phase

After the sender node receives messages, they should be reconstituted to the original form.

4.3.1. CAN Message Decryption Phase

The resulted message is decrypted to obtain the original message.

4.3.1. Calculating Clock Cycle

The last step of our methodology is calculating the clock cycle needed to perform a CAN data transmission (more detailed information can be found in the second sub-section of the next section).

5. DESIGN OF INTRUSION DETECTION SYSTEM

Our goal in this paper is to design an IDS aims at detecting attacks on CAN bus network and based on analyzing of message frame. To reach our goal we adopt three steps: analyze CAN messages, inject malicious frames on bus network and implement the proposed algorithm. This section details the design of this IDS: Subsection A describes the system model, in subsection B, we give the threaten model and the third subsection details the fundamental idea of our approach.

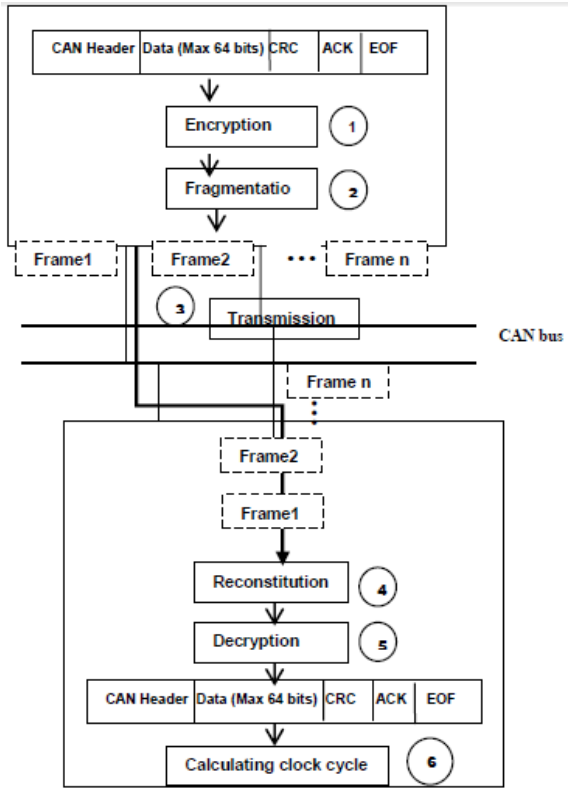

Figure 3. Overall process of analysis Methodology for a Secure CAN Bus Communication

5.1. System Model

We adapt the automotive network architecture consists of three nodes connected to CAN bus in the vehicle via a serial data communication bus. Each ECU controls a particular function of the vehicular system. As shown in Figure 4.

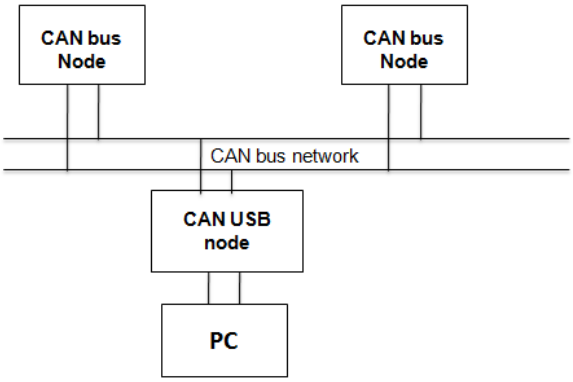

Figure 4. System Model

5.2. Threaten Model

If we consider the way over it the attacker could accede to the network, we assume this attack models: the adversary access to the CAN network by compromising an existing ECU. The attacker compromise one ECU to send frames with correct ID but from different ECU compared to its legitimate one. So, authorized ECUs believe that is a legitimate message and it is sent by an authorized one. The goal of our work is the detection of intrusions regardless to their origins.

As mentioned later, CAN bus has not any type of authentication. Therefore, if an attacker succeeds to access to the bus, he could get code running on an ECU (via an attack over Bluetooth, telemetric, tire sensor, physical access…). Also, she could full control the vehicle by injecting spoofed messages. As the attacker tries to send a malicious message to an ECU, as well as authorized ECUs still send their normal messages periodically. So, the target ECU will receive messages from the authorized ECU and from the attacker. Thus, the attacker reaches his goal to transmit injected message, unless she sends it faster than the original ECU. Previous researches like [19] and [22] mentioned that an attacker should send messages from 20-100 times faster than the original ECU to make the target ECU listens to the injected messages. Finally, the rate of messages on the network will be increased more than two times (20 – 100 times) higher than the normal

5.3. The Fundamental Idea

In the following subsection, we describe the different features of our IDS, as well as the working process.

5.3.1. IDS Description

We adapt the Intrusion detection system with this aspect:

- Data source: the proposed IDS is a network intrusion detection systems (i.e.it analyzes incoming network traffic).
- Method of detection: as its ability to detect new attack as well as its easy implementation than the Signature- based IDS, we adapt Anomaly-based IDS.
- Frequency of analysis: the detection is in real time
- Concerning its behavior after detection, our IDS is dedicated to alert the user if suspicious frame is detected
- In our work the IDS dedicated to detect frames including incorrect ID and malicious frames generated periodically while the transmission of a normal traffic.
- As each authorized ECUs send their normal messages periodically, the time interval of each CAN ID is unique. Therefore, our IDS detects messages which their IDs do not respect their own interval time, as the procedure in the next section.

5.3.2. Process Principal

After introducing the main aspects of our IDS, we continue with presenting the procedure according to it our system detects messages: each ECU connected to CAN bus sends its message regularly. So, each message ID (0x1, 0x2 ,…) has its own regular frequency or interval. The IDS checks the arrival time of CAN ID. It calculates the time interval of the arrival message compared to last

message. If the interval message is less than the normal one, the alert will be lifted. The entire process of the IDS is summarized in Figure 5.

6. TEST ENVIRONMENT

This section is dictated to detail the test environment of the proposed IDS: we give the hardware architecture in subsection A. Subsection B describes the experimental setup. Last subsection details algorithms process of the analysis method.

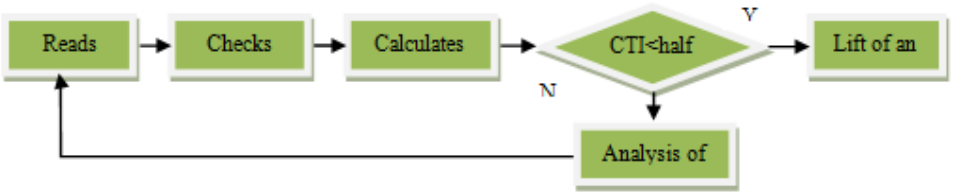

Figure 5. Flowchart of the proposed intrusion detection system

6.1. Hardware Architecture

In Figure 6, the block diagram of the proposed analysis methodology is shown (system 1). This system consists essentially of two CAN nodes which are all connected to a transmission medium (medium) looped by two termination resistors.

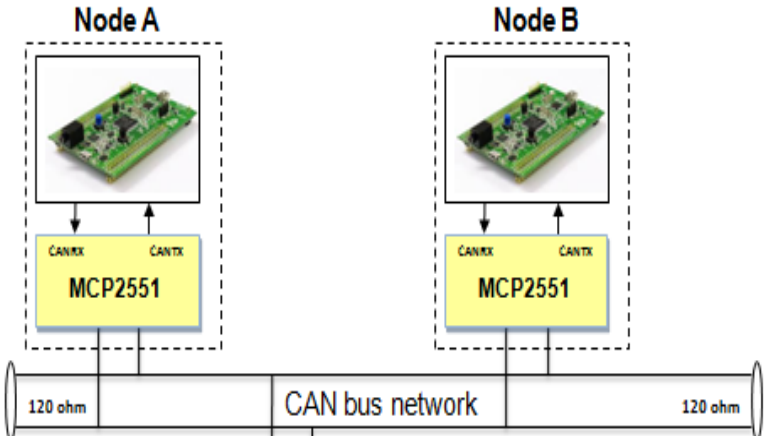

Figure 6. Block diagram of System1

The block diagram of the proposed IDS (system 2) is shown in Figure 7.This system consists essentially of two CAN nodes and a CAN-USB node which is all connected to a transmission medium (medium) looped by two terminating resistors.

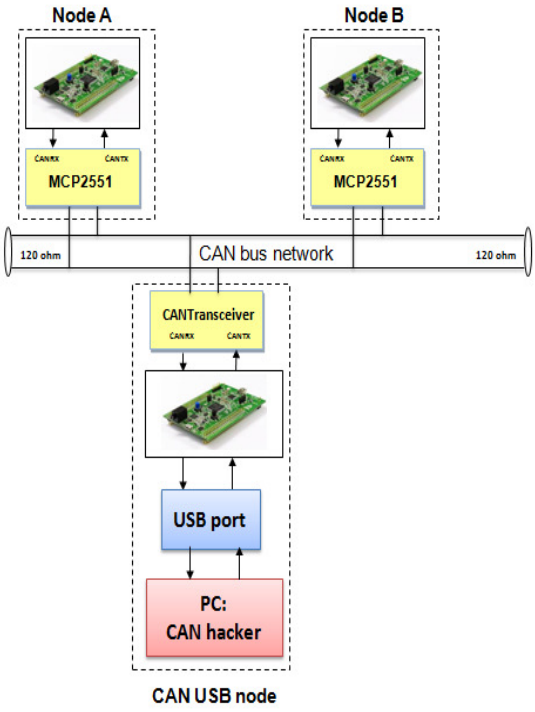

Figure 7. Block diagram of System2

6.2. Description of Platform

To design our methodology which allows implementing cryptography protocols and IDS, we proposed a platform compound of CAN nodes as shown in Figure 8. We chose ST Micro electronics'32F407 microcontroller board [23] with a 32 bit ARM Cortex-M4 core clocked at 16 MHz and an adaptive real-time accelerator since it is characterized by features could help as in our application. As CAN transceiver, we chose MCP2551 .We made the transmission in twice to remove any type of parasite terminal by two 120W resistor to provide CAN bus communication capabilities. For the implementation of algorithms, we proposed the Keil MDK 5 as an integrated development environment (IDE) to program STM32 and the STM32CUBEMX tool for configuration.

To adopt security into the CAN bus network, we included the STM32 cryptographic library package (X-CUBE CRYPTOLIB) in particular AES-128 in CMAC mode which has a 128-bit long key and a 128-bit message in output. Since the CAN data message could contain 108 bits in totally, we choose to append MAC to the data field and truncated it to 2 bytes then concatenated the result after be decrypted. The AES-CMAC library is taken from the [24]. The total code size for AES computation was about 2 244 bytes and CAN communication was about 2500 bytes. AES modes CMAC do not have a proper decryption mode like ARC4. So, decryption works exactly like encryption. The tests were executed on STMF4 which their CPU is running at 168MHz. The number of cycles needed is calculated according to equation (2) depending on [25].

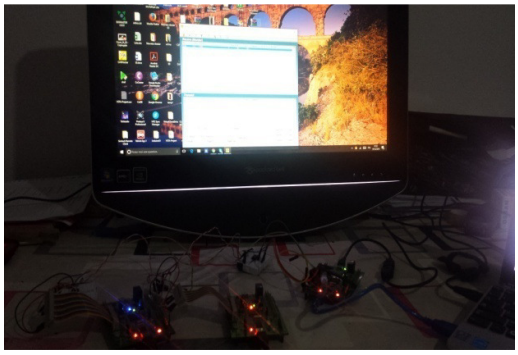

Figure 8. Components of experimental setup

6.2. Algorithms Process of the Analysis Method

In this subsection, we give flow diagrams of different phases

Main Program

The Figure 9 above shows the flow diagram of the main program. At the beginning, we start by the configuration of the stm32. Then we initialize the different devices to use such as USB, CAN and ADC ... Finally, the program is ended by the activation of interruptions if they are triggered.

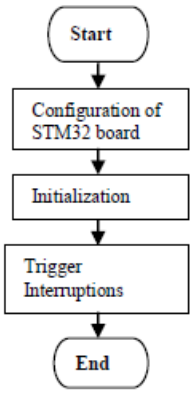

Figure 9. Flow diagram of the main program

AES-CMAC Encryption Algorithm

The following flowchart describes the AES_CMAC Encryption algorithm. Referring to Figure 10, the sender starts the initialization for AES-CMAC Encryption and checks the error status. If the error statutes value is “AES_ERR_BAD_CONTEXT” and “AES_ERR_BAD_ PARAMETER”, the sender node ends the process. Else if the error statutes value is” AES_ SUCCESS”, the sender encrypts data in CMAC Mode and checks the error status. If the error statutes value is “AES_ERR_BAD_PARAMETER”, “AES_ERR_BAD_OPERATION” and “AES_ERR_BAD_INPUT_SIZE”, the sender node ends the process. Else if the error statutes value is” AES_SUCCESS”, the sender finalizes of CMAC Mode and checks the error status. In the both value of the error statutes the sender node ends the AES encryption process.

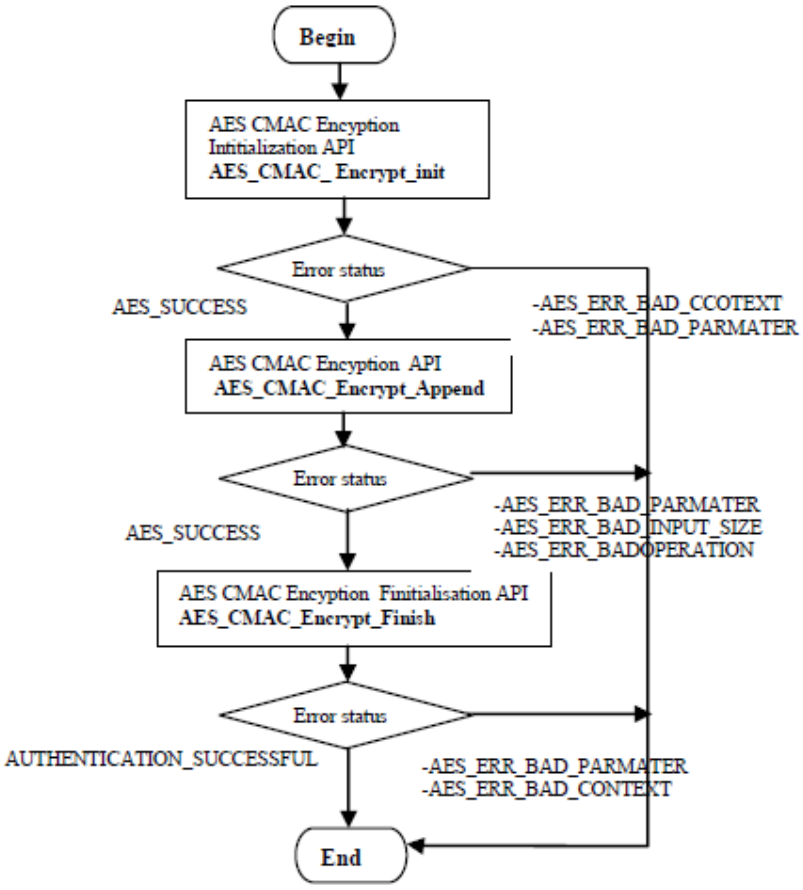

Figure 10. Flow diagram of AES_CMAC algorithm Encryption

AES-CMAC Decryption Algorithm

The next flow diagram describes the AES_CMAC Decryption algorithm. Referring to Figure 11, the receiver starts the initialization for AES-CMAC Decryption and checks the error status. If the error statutes value is "AES_ERR_BAD_CONTEXT" and "AES_ERR_BAD_PARAMETER", the receiver node ends the process. Else if the error statutes value is" AES_SUCCESS", the receiver decrypts data in CMAC Mode and checks the error status. If the error statutes value is "AES_ERR_BAD_PARAMETER","AES_ERR_BAD_OPERATION" and "AES_ERR_BAD_ INPUT_SIZE", the receiver node ends the process. Else if the error statutes value is" AES_SUCCESS", the receiver finalizes of CMAC Mode and checks the error status. In the both value of the error statutes the sender node ends the AES decryption process.

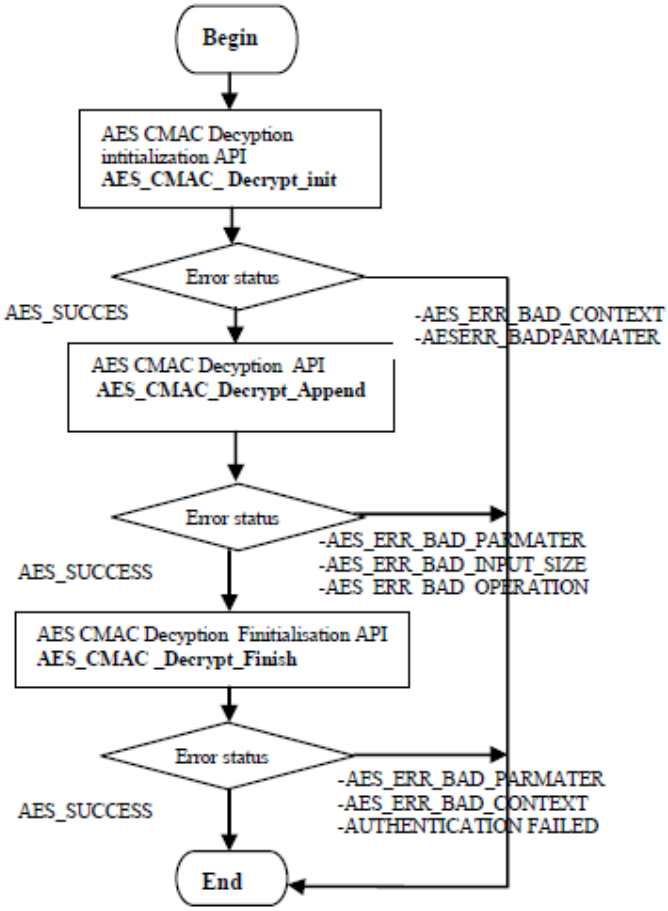

Figure 11.flow diagram of AES_CMAC algorithm Decryption

To determine the impact of using a cryptographic algorithm in CAN bus communication, we need to calculate the number of cycles clock transmit a CAN data. At first we have the number of cycles needed to perform each process is defined as follows:

$$Cycles = Init\ key\ cycle + Init\ message\ cycle + Process\ block\ of\ data\ cycle * number\ of\ blocks \quad (1)$$

So the number of cycles needed to perform a CAN data transmission is calculated as follow

$$Cycles_{CAN} = Init\ key\ cycle + Init\ message\ cycle + Process\ block\ of\ data\ cycle * number\ of\ blocks + 2*(min\ of\ CAN\ data\ transmission\ cycle) \quad (2)$$

7. CONCLUSION

Our main contribution in this paper was the design of a tool that allows on the one hand the calculation of real-time performances resulting from the implementation of cryptographic mechanisms. On the other hand, the proposed system is dedicated to implementing an intrusion detection mechanism for CAN networks that we have designed. The method is based on the analysis of the time intervals of the CAN message. Also, in this work we have developed an

efficient experimental platform for the analysis, the implementation of a secure communication on the CAN bus and the injection of the usurped messages. .As perspective of this work, we intend to evaluate proposed methods by the implantation and the comparison between them.

REFERENCES

[1] C. Miller, C. Valasek, (2015) “Remote exploitation of an unaltered passenger vehicle”, BlackHat USA.

[2] R. N. Charette, (2009) “This car runs on code,” IEEE Spectr.,vol. 46, no. 3, p. 3.

[3] R.B.GMBH, (2014) “Bosch Automotive Electrics and Automotive Electronics”, 5 ed. Bosch Professional Automotive Information. Springer Vieweg.

[3] TEXAS INTRUMENTS, (2016) “Introduction to the Controller Area Network (CAN)”, Application Report, SLOA101B–August 2002–Revised May 2016.

[4] C-W. Lin, A. Sangiovanni-Vincentelli, (2012) “Cyber Security for the Controller Area Network (CAN) Communication Protocol”.

[5] S. Checkoway, D. McCoy, et al., (2011) ”Comprehensive experimental analyses of automotive attack surfaces”, Proc.20th USENIX Security, San Francisco, CA.

[6] C. Miller, C. Valzek, (2013)“Advanture in automotive networks and control units”.

[7] K. Koscher, A. Czeskis, et al., (2010) “Experimental Security Analysis of a Modern Automobile, IEEE Symposium on Security and Privacy”.

[8] Prescott.E.Small, (2011) “Defense in Depth: An Impractical Strategy for a Cyber World"

[9] M. Wolf, A. Weimerskirch, & C. Paar, (2004) “Security in automotive bus systems“, Workshop on Embedded Security in Cars.

[10] D.K. Nilsson, U.E. Larson, and E. Jonsson, (2008) “Efficient In Vehcile Authentication Codes”, Vehicular Technology Conference VTC.

[11] A. Van Herrewege, D. Singelee, I. Verbauwhede, (2011)“Canauth - a simple, backward compatible broadcast authentication protocol for can bus”, ECRYPT workshop on Lightweight Cryptography.

[12] B.Groza, S.Murvay, et al., (2012)“LiBrA-CAN: A Lightweight Broadcast Authentication Protocol for Controller Area Networks”, International Conference on Cryptology and Network Security CANS 2012: Cryptology and Network Security pp 185-200.

[13] S.Woo, H.J.Jo, and D.H.Lee, (2014) “A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN “, In IEEE Transactions On Intelligent Transportation Systems.

[14] S.Nurnberger and Ch. Rossow,(2016) “VatiCAN -Vetted, Authenticated CAN Bus“, International Conference on Cryptographic Hardware and Embedded Systems(CHES)’2016.

[15] M. Müter, N. Asaj, (2011)“Entropy-based anomaly detection for in-vehicle networks”, Intelligent Vehicles Symposium (IV), Baden Baden, Germany, IEEE.

[16] T. Hoppe, S. Kiltz, and J. Dittmann, (2009) “Applying Intrusion Detection to Automotive IT - Early Insights and Remaining Challenge“, Journal of Information Assurance and Security (JIAS), pp. 226-235.

[17] U. E. Larson, D. K. Nilsson, and E. Jonsson, (2008)" An Approach to Specification-based Attack Detection for In-Vehicle Networks".

[18] C.Miller and C.Valasek, (2014)"A survey of remote automotive attack surfaces". Last Accessed from http ://illmatics.com/remote attack surfaces.pdf.

[19] T. Matsumoto, M. Hata, et al., (2012)"A method of preventing unauthorized data transmission in controller area network", Vehicular Technology Conference (VTC Spring), pages 1–5, Yokohama, Japan, IEEE

[20] I. Studnia, E. Alata, et al., (2015)"A language-based intrusion detection approach for automotive embedded networks", The 21st IEEE Paci_c Rim International Symposium on Dependable Computing (PRDC 2015), Nov 2014, Zhangjiajie, China. Proceedings of the 21st IEEE Paci_c Rim International Symposium on Dependable Computing (PRDC 2015).

[21] H. M. Song, H. R. Kim and H. K. Kim, (2016)"Intrusion Detection System Based on the Analysis of Time Intervals of CAN Messages for In- Vehicle Network".

[22] (2017) "STM32F407VG", http://www.st.com/en/microcontrollers/stm32f407vg.html.

[23] (2015) "UM1924 User manual STM32 crypto library", www.st.com.

[24] (2013) "UM0586 User manual STM32 Cryptographic Library", www.st.com.

Authors' Information

[1]National Engineers school of Gabes ENIG, Tunisia, Avenue Omar Ibn Alkhattab, Zrig Gabes 6029

[2,3] Engeneering School of Sfax ENIS Tunisia, Route de Soukra, Km 3.5 BP W, 3038 Sfax

[1]Mabrouka Gmiden received the engineering degree in electric and automatic from the national Engineers school of Gabes (ENIG), Tunis in 2012. She is currently working toward the Ph.D. degree at National Engineers school of Gabes (ENIG). She is a member of Computer Embedded Systems Laboratory (CES Lab) in the national Engineers school of Sfax (ENIS). His research interests include automotive, security, cryptography, CAN bus security. E-mail: mabroukagmiden@hotmail.fr

[2]Mohamed Hedi Gmiden received the B.S. degree in Electrical Engineering, the M.S degree and Ph.D. degree in automatic and industrial Computing from in the national Engineers school of Sfax (ENIS), University of Sfax, in 1996, 2004 and 2011 respectively. He joined the Tunisian University in 2007. He is currently an assistant professor in Higher Institute of Industrial Systems (ISSIG), University of Gabes. He is a member of Computer Embedded Systems Laboratory (CES Lab).
E-mail: mohamedhedi.gmiden@enis.rnu.tn

[3]**Trabelsi Hafdh** received the B.S. degree from Sfax Engineering School (ENIS), University of Sfax, Sfax, Tunisia, in 1989, the M.S. degree in the Central School of Lyon, France, in 1990, the Ph.D. degree from the University of Paris XI Orsay, France, in 1994, and the "Habilitation Universitaire" in the National Engineering School of Sfax (ENIS), University of Sfax, Sfax, Tunisia, in 2008, all in electrical engineering. He is working toward the Research Management Ability degree in the field of electrical machine design at SES. He joined the Tunisian University, Tunisia, in 1995. He moved to the ENIS in 2000. He is currently a Professor of Electrical Engineering. He is a member of the Research Unit on Renewable Energies and Electric Vehicles of the University of Sfax, where he is the chair of the Electric Machine Design Team. He is a member of the Organizing Committee of the IEEE International Conference on Signals Systems Decision and Information Technology. E-mail: hafedh.trabelsi@enis.rnu.tn