Avoidance of Replay attack in CAN protocol using Authenticated Encryption

Selvamani Chandrasekaran[1]
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore, Amrita
Vishwa Vidyapeetham, India
selva.vinoth205@gmail.com

K I Ramachandran[2]
Centre for Computational Engineering and Networking
Amrita School of Engineering, Coimbatore, Amrita
Vishwa Vidyapeetham, India
ki_ram@cb.amrita.edu

Adarsh S[3]
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore, Amrita
Vishwa Vidyapeetham, India
s_adarsh@cb.amrita.edu

Ashish Kumar Puranik[4]
Tata Consultancy Services,
Pune, India
ashishk.puranik@tcs.com

Abstract - Controller Area Network is the prominent communication protocol in automotive systems. Its salient features of arbitration, message filtering, error detection, data consistency and fault confinement provide robust and reliable architecture. Despite of this, it lacks security features and is vulnerable to many attacks. One of the common attacks over the CAN communication is the replay attack. It can happen even after the implementation of encryption or authentication. This paper proposes a methodology of supressing the replay attacks by implementing authenticated encryption embedded with timestamp and pre-shared initialisation vector as a primary key. The major advantage of this system is its flexibility and configurability nature where in each layer can be chosen with the help of cryptographic algorithms to up to the entire size of the keys.

Keywords – CAN protocol, Vulnerabilities, Replay Attacks, Spoofing attacks, Tiny Encryption Algorithm (TEA), Advanced Encryption Standard (AES), Message Authentication Algorithm (MAC), Secure Hash Algorithm (SHA), Timestamp, Counter, Authenticated Encryption.

I. INTRODUCTION

The world is drifting towards connected cars and autonomous vehicles. By 2030 at least 250 million connected cars and 10 million self-driving cars are expected to be on road. The automotive industry has evolved in various sectors from the day of invention of IC engines [1][2]. The number of mechanical parts has reduced to sensors and actuators. This increases the efficiency, comfort, compactness and safety of the vehicles [3]. The language of communication between the systems are established between different communication protocols [4]. These communication protocols play a vital role in connected and autonomous vehicles. When the number of devices connected in the system increases, security threats also increases substantially. Cyber security plays an indispensable role in addressing the threats and vulnerabilities in the system [5].

One of the primary communication protocols in automotive industry is CAN protocol. [6][7]. Controller area network (CAN) is de facto standard in vehicular networks from 1993. It is a message-based protocol in which every time the node is triggered to send a high priority message which wins arbitration and the message gets communicated. It is also a multi- master bus network where every node act as a master as well as slave. CAN protocol interacts with three layers out of seven in ISO /OSI model standard. (Physical layer, Data link layer and Application layer).

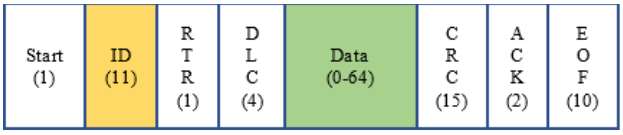

Fig.1 Standard CAN Bus Frame.

A standard CAN frame shown above consists of a single bit start frame for synchronization followed by 11-bit identifier. The difference between standard CAN frame and extended CAN frame is number of bits in the identifier. Extended CAN frame uses 29 bits as identifier [8][9]. The single bit RTR (Remote Transmission Request) bit is used for request /respond status of nodes in the frame. The length of the data is specified in the DLC (Data Length Control). The data block consists of actual data of transmission with (0-64 bits) in normal messages. Along with that CRC (Cyclic Redundancy Check) of 15 bits and ACK (Acknowledge) of 1 bit is used for data integrity and consistency. The frames are terminated by 7-10 bits of EOF (End of Frame) block to acknowledge initiation of the next frame in the bus [10].

CAN protocol is robust with its salient features like arbitration, message filtering, error detection, data integrity, data consistency and fault confinement [11].

II. LACK OF SECURITY IN CAN NETWORKS:

Despite of its robust and reliable architecture CAN networks lacks in security aspects which leads to highly vulnerable attacks [6] [14]. Some of the security gaps are as listed below -

A. Lack of authentication:

CAN protocol lacks in provision for any type of message authentication or node authentication mechanism to recognize a sender of specific message or identification of a node. The protocol is inadequate to distinguish between a legitimate and an illegitimate (malicious) node. The trust ability of the nodes is highly compromised due to lack of authentication.

B. Lack of addressing:

Since CAN is a message-based protocol it doesn't have any node addresses. Nodes without specific address pursues high time risk of transmitting malicious message. Identification of the source of a message is difficult in case of CAN networks without node address [6] [17].

C. Lack of Encryption:

One of the major vulnerabilities in CAN architecture is the absence of encryption by its nature of design. It lacks to provide any type of encryption mechanisms to ensure the data confidentiality and integrity during transmission [18].

D. Denial of Service Attack:

Although arbitration mechanism in CAN protocol is used to transmit time critical and high priority messages effectively, it paves a way to flood the network with denial of service attack [19] [20]. The continuous transmission of high priority messages from an authorized / unauthorized node in the bus will lead the system not to listen to any other messages transmitted by the actual critical node. By this, actual service of the nodes in the system would get denied. This continuous flooding of high priority messages in the bus occupies the total bandwidth of the bus and downs the network. This is referred as Denial of Service attack (DoS) in the CAN networks.[21]

III. REPLAY ATTACKS

Apart from the above vulnerabilities, one of the prominent and highly vulnerable attack that can be easily made on the CAN bus networks is replay attack. [22] It is considered as highly critical because it can be performed even when the messages are encrypted. This paper focuses on the impact of replay attack over CAN networks and the suppression methodology on it.

Replay attack is a simple and efficient Man in The Middle (MITM) attack where the messages are replayed or retransmitted continuously or delayed in time. This replay attacks are otherwise called as "record and playback attacks" or "cut and paste attacks" [23]. This replay attack is considered to be dominant because it can also be performed in the secure environments. It can replay or retransmit the encrypted messages [24].

This replay attack causes adverse effects on the vehicle environment. For instance, once the brake is applied in the car by the driver, the braking system sends the brake message to the other corresponding nodes with the help of CAN bus network [25]. An adversary in the network can easily capture the message transmitted by braking unit and save for later purpose. In case the adversary wants to crash the car at high speed he can program to retransmit the message which causes fatal accidents as depicted in the following figure.

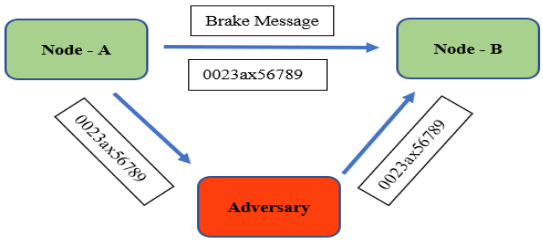

Fig.2 Replay attack scenario.

IV. THREAT MODEL AND EXPERIMENTAL SETUP FOR REPLAY ATTACK.

This setup is to showcase the strategies of replay attack and the means of suppressing counter measure model over CAN network. It exhibits a typical model of collision avoidance system with a distance sensor and a controlling unit.

A. Method of Spoofing:

The experimental setup is of a typical model of collision avoidance system with distance sensor (M16 – Solid State LiDAR) and control unit with microcontroller (LPC -1768) and an adversary (Raspberry Pi powered with Kali Linux Operating System and connected with MCP 2515 CAN transceiver). For instance, let us assume that the control unit is programmed with collision avoidance in the range of 30 meters. The control unit will send signal to the braking system when the object range is below 30 meters.

In case of this setup an intruder / adversary connected to the same network can read the messages in the network. The adversary node records all the distance messages transmitted via CAN bus to the LPC 1768 unit. The typical message record scenario in the setup is shown in the below figure.

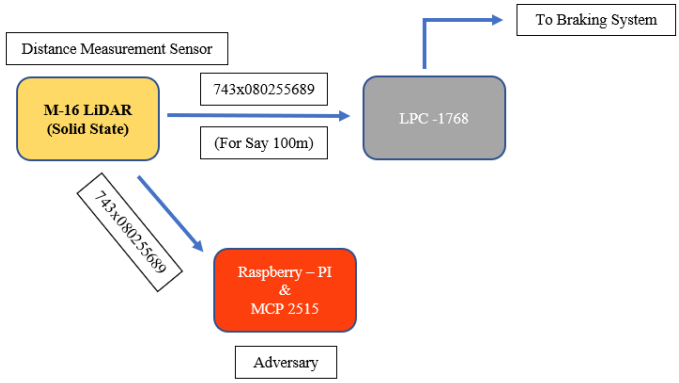

Fig.3 Record Transmit Message Scenario

When the messages are recorded it can be played back with the delay or the time required. Considering the scenario, the distance measurement sensor (M-16 Solid State LiDAR) transmits the distance 100 meters when the object is far from the sensing range. When the vehicle starts approaching towards the object the distance reduces and it falls below 30 meters and brake actuation should happen to avoid collision. But if an adversary records the distance messages at 100 meters and executes the same while actual distance is below 30 meters. The control unit recognizes it as actual message and fails to send signals to braking unit which leads to collision of vehicle. This type of record and playback

attack typically termed as replay attack can be implemented in multiple critical scenarios. The playback scenario block diagram is shown below.

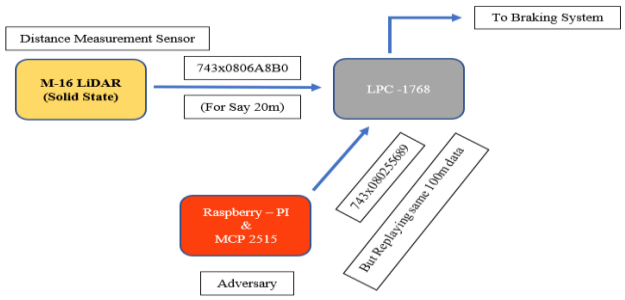

Fig.4 Replay / Playback Message Scenario

The setup of spoofing of CAN messages is done and analyzed to record playback conditions in real-time scenario. The photograph of the system setup is given in figure below. The actual setup where the distance sensor output is shown as beams in the laptop screen and the corresponding data is shown in the first window of the raspberry pi monitor. The second window plays a replay attack where the system responds to the adversary and fails the actual condition and lets the service down.

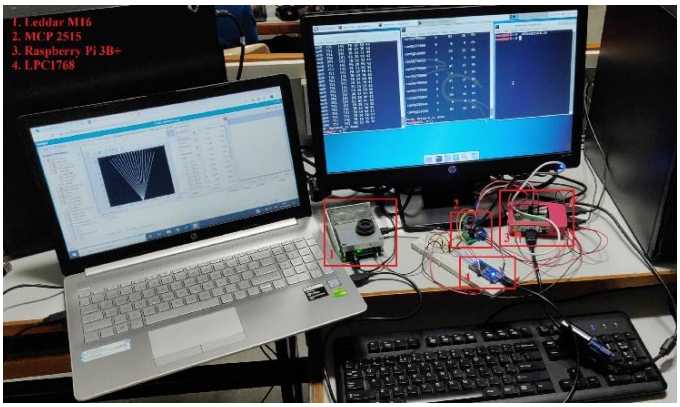

Fig.5 Working Setup with Record /Playback scenarios

B. Components Used:

1) M16 - Leddar Sensor Module:

LEDDAR stands for Light-Emitting Diode Detection and Ranging is used to detect, locate and estimate the distance between the target objects. It communicates through standard CAN protocol. The device used in our setup is of 16 channel solid state LiDAR which is shown below.

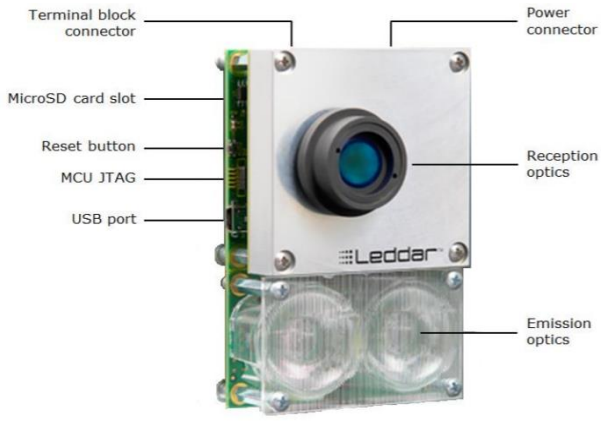

Fig.6 M16 - Leddar Sensor Module

2) LPC1768

The microcontroller LPC1768 is NXP based ARM M3 Cortex core with 96 MHz, 64 KB RAM, 512 KB flash, and has interfaces inclusion of , CAN, SPI, I2C ,USB Device and other I/O. Figure 7 shows pinout of the LPC1768 controller.

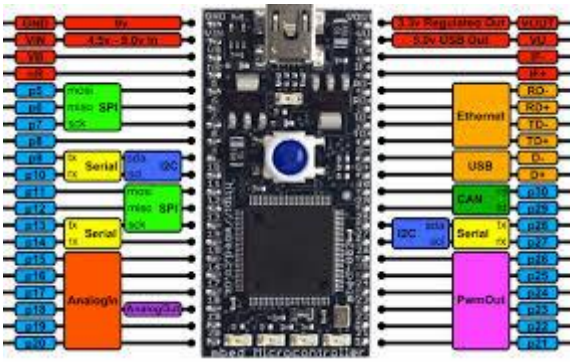

Fig.7 Pinouts of LPC – 1768

3) Raspberry Pi

The Raspberry Pi is a mini computer is powered with Kali Linux to give additional libraries to perform message sniffing and replaying. The pinout details of Raspberry Pi 3B+ is given in the figure 8.

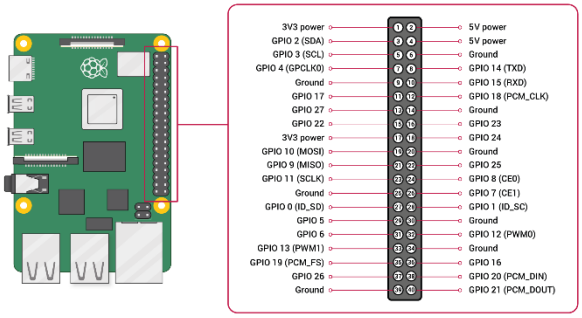

Fig.8 Raspberry Pi 3B + pinouts.

4) MCP2515

MCP 2515 is a low cost can controller which has Serial Peripheral Interface (SPI) to connect with Raspberry Pi. Fig. 9 provides the pin out details of MCP-2515.

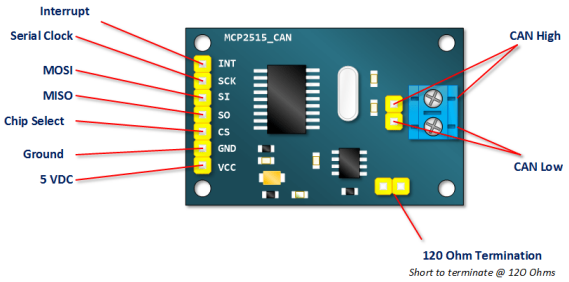

Fig.9 Pinouts of MCP2515

V. SUPRESSION METHODOLOGY AND COUNTER MEASURES OF REPLAY ATTACK

The critical part of replay attack is that it can happen to completely secured and encrypted system. Some of the methods of supression is authenticating or encrypting the messages using unique values. Assigning a counter or timestamp along with encryption ensures the uniqueness of message. Both the systems perform well with replay attacks and it has its inherent pros and cons.

The replay attack can be supressed by identifying the originality of the message. Implementing counter in the transmission side and verifying it in the reception ensures the correctness of the message. But the weakness of this system is it works well with node to node communication and becomes tedious in intialising and updating the counter values with increase in number of nodes.

The next widely accepted method is timestamp synchronisation with the messages. It synchronises the global timestamp along the nodes which ensures the freshness of message at every instance of time. This paper incorporates timestamp method as part of the algorithm methodology.

Prior to inclusive of supression technique towards replay attack, preliminary security should be ensured to the system. There are several methods implemented along CAN network in terms of authenticity and confidentiality of messages in the network. Numerous reseraches have been conducted on cryptographic algorithms used and implementation of security policies over CAN systems. This paper proposes a novel approach of using authenticated encryption which ensures both authenticity or confidentiality of messages. The traditional system uses either one of it.

A. Authenticated Encryption

Authenticated encryption assimilates flavours of both authentication and encryption mechanisms. So this protocols provides both authenticity and confidentiality of data.

B. Authentication:

Authentication is a mechanism to check authenticity and integrity of the data. In CAN networks authentication can be applied in two ways, one is via node authentication model and the other one is message authentication model. Node authentication model ensures the trustability of nodes where legitimate nodes and illegitimates nodes are authorised by a centralised server by validating physical addresses of the nodes or key exchange between the nodes.

The other one is message authentication model which encompasses message authentication code generally termed as MAC. In this technique, the integrity of the messages are validated by different message authentication code (MAC) algorithms based on the key size. Message authentication code (MAC) is a model where the tag is attached to a message to ensure the message integrity and message authenticity, as shown in Fig. 10. The tag is derived by applying a MAC algorithm to message with a secret key.

Fig.10 Message Authentication Code

Although authentication mechanism ensures integrity and authenticity it equally has a drawback of hash collision attack or widely termed as birthday paradox.

C. Encryption:

The concept of encryption is universally known. The process of ciphering data by a cryptographic key and transmitting into the network / bus and decrypting it at the receiver side to get the original message. It corroborates the confidentiality of the data in the network medium and avoids Man in the Middle (MITM) attack. Figure 11 illustrates the encryption and decryption.

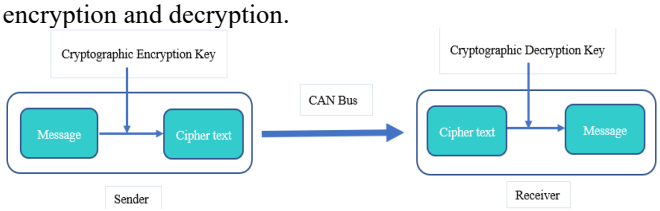

Fig.11 Encryption and Decryption

Same like authentication, encryption has a disadvantage known as chosen plaintext attack.

To overcome drawbacks of both authentication (hash collision) and encryption (chosen plain text attack) blend of both used as authenticated encryption.

VI. PROPOSED METHODOLOGY OF IMPLEMENTATION

The whole approach is divided into two logical sections. Both encryption and decryption modules will be in each and every single node. For methodological explanation in this experiment, one node will be acting as transmitting encryption section and the other will be acting as receiving decrypting section.

The transmitter node consists of three modules; one is of primary key generation where 6 bits of initial key vector and 10 bits of current time stamp combined to form 16 bits of primary key. The key length can be varied with respect to the cryptographic algorithms used. The other section consists of the encryption module where it encrypts the CAN message along with the primary key generated. Tiny encryption algorithm (TEA) is used for encryption here for light weight condition. The third is of generating the hash code of the primary key which is associated with the current time stamp. Both hash code and cipher text are transmitted to the receiver via CAN bus. The encryption block diagram is shown in Figure 12.

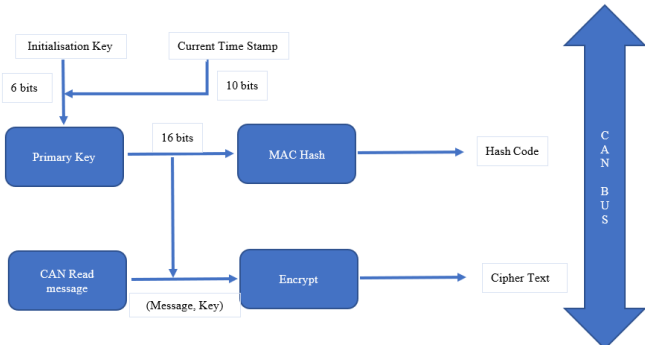

Fig.12 Authenticated Encryption – Transmitter Node

Same like transmitter node, receiver node consists of modules like primary key generation module, MAC generation module, hash comparison module and decryption module. The hash generated along with Initial Primary Key and Time Stamp combination is compared with hash received from the transmitter via bus. If both hash codes are equals, then the cipher text is decrypted to actual message. The decryption block diagram is shown in Fig.13.

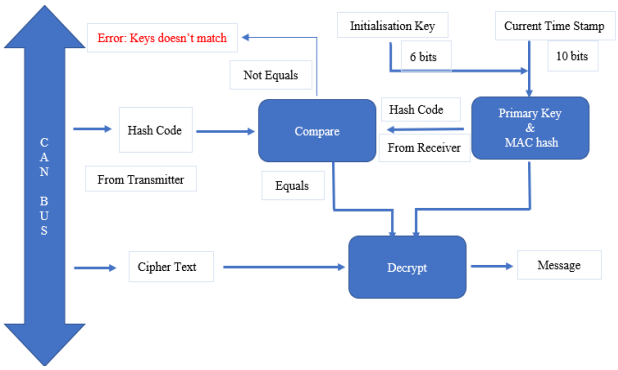

Fig.13 Authenticated Decryption – Receiver Node

VII.RESULTS AND DISCUSSION

The counter measure programming is done in Python programming language with Kali Linux powered in Raspberry Pi. The results of the encrypted and decrypted messages and replay attack failure conditions are shown in Figure 14 and Figure 15.

A. Encryption module:

The encryption module results the Timestamp instance of encryption, actual CAN message, hashtag, generated and the cipher text.

```
selva@kali:~/Documents/Python/Demo$ python3 canread.py
CAN Message from Node : 0x3e0x40x2b0x26
Time_Stamp : 17:05:2020-14:00:04
selva@kali:~/Documents/Python/Demo$ python3 encryptmsg.py
The time stamped key: abcdef0517201400
Time_Stamp : 17:05:2020-14:00:14
The CAN Message 0x3e0x40x2b0x26
The length of mesaage  15
The Encrypted message YDLcUm+jBZQj+A8l7nieXw==
The length of mesaage  24
The hash tag generated with time stamp key: 2f4f017b8dca72801d157457d3b5e1c7cd7e44e6
The length of the tag  40
selva@kali:~/Documents/Python/Demo$
```

Fig.14 Authenticated Encryption Scenario

B. Decryption module:

Similarly, the decryption module in the receiver results with Time Stamp, Cipher text received from CAN Bus, hashtag received from CAN Bus and actual message after decryption.

```
selva@kali:~/Documents/Python/Demo$ python3 decryptmsg.py
Time_Stamp : 17:05:2020-14:00:43
Encrypted message YDLcUm+jBZQj+A8l7nieXw==
The encryption hash tag: 2f4f017b8dca72801d157457d3b5e1c7cd7e44e6
The hash tag generated with time stamp key at decryption side:  2f4f017b8dca72801d157457d3b5e1c7cd7e44e6
The decrypted original message : 0x3e0x40x2b0x26
The length of mesaage  15
selva@kali:~/Documents/Python/Demo$
```

Fig.15 Authenticated Decryption Scenario

C. Replay Attack Failure:

The decryption module works only when the time stamp hash generated is equal to the hash received from the CAN bus. In case the data generated or transmitted is different from the time of receptions. The keys for decryption won't be matched. So, every time only fresh messages will be decrypted. So, the chances of playing back the same message won't be decrypted at any cost. So only fresh messages associated with right key will be decrypted. The replay attach failure scenario is shown in Fig. 16.

```
selva@kali:~/Documents/Python/Demo$ python3 decryptmsg.py
Time_Stamp : 17:05:2020-14:01:30
Encrypted message YDLcUm+jBZQj+A8l7nieXw==
The encryption hash tag: 2f4f017b8dca72801d157457d3b5e1c7cd7e44e6
The hash tag generated with time stamp key at decryption side:  7f2670c0f1338ef10a51f8a82645b1a65f3ef92a
The keys doesnt match or expired
selva@kali:~/Documents/Python/Demo$
```

Fig.16 Replay Attack Failure Scenario

VIII.CONCLUSION

The advantage of suppression technique used in this paper is to avoid replay attack i.e. configurable to any layer with respect to system conditions and user requirements. The cryptographic algorithms used here such as Tiny Encryption Algorithm (TEA) for encryption or SHA (Secure Hash algorithm) for authentication can be changed to any required algorithms such as AES or SPECK. The 16-bit key length used here also, can be changed to either 32-bits or 64-bits. The substantial merit over this system is that it uses same key for the encryption as well as hashing. For system simplicity Initialization vector key is a pre-shared key. Finally, the configurable elapse time for hash tag validation, results in playback control and packet sniffing is considered to be a gem in the crown for this method.

REFERENCES

[1] Bosch, R. CAN Specification Version 2.0, CAN, 1991.

[2] Bosch, R. CAN with Flexible Data Rate Specification Version 1.0, CAN, 2012.

[3] CSS Electronics. CAN Bus Explained; CSS Electronics: Aarhus, DK, Jutland, 2019.

[4] Hoppe, T, Dittman, J. Sniffing/Replay Attacks on CAN Buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy. In Proceedings of the 2nd workshop on embedded systems security (WESS), Brussels, Belgium, 4 October 2007 pp. 1–6.

[5] Rebecca, B. Proof-of-concept CarShark software hacks car computers, shutting down brakes, engines, and more. Popular Science. Available online: https://www.popsci.com/cars/article/2010-05/researchers-hack-carcomputers-shutting-down-brakes-engine-and-more (accessed on 29 May 2018).

[6] Siddiqui, Ali Shuja & Gui, Yutian & Plusquellic, J. & Saqib, Fareena. (2017). Secure communication over CANBus. 1264-1267. 10.1109/MWSCAS.2017.8053160.

[7] P. Noureldeen, M. A. Azer, A. Refaat and M. Alam, "Replay attack on lightweight CAN authentication protocol," 2017 12th International Conference on Computer Engineering and Systems (ICCES), Cairo, 2017, pp. 600-606, doi: 10.1109/ICCES.2017.8275376.

[8] Ankit Shah(2018) . A survey of cryptographic algorithms for IoT devices . Proceedings of ICSICCS-2018.

[9] Radu, Andreea & Garcia, Flavio. (2016). LeiA: A Lightweight Authentication Protocol for CAN. 9879. 283-300. 10.1007/978-3-319-45741-3_15.

[10] J. Halabi and H. Artail, "A Lightweight Synchronous Cryptographic Hash Chain Solution to Securing the Vehicle CAN bus," 2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), Beirut, 2018, pp. 1-6, doi: 10.1109/IMCET.2018.8603057.

[11] ICS-CERT (2017). CAN Bus Standard Vulnerability. [online] ICS-CERT, p.1. Available at: https://ics-cert.us-cert.gov/alerts/ICS-ALERT-17-20901 [Accessed 25 Mar. 2018].

[12] Jukl, M. & Čupera, Jiří. (2016). Using of tiny encryption algorithm in CAN-Bus communication. Research in Agricultural Engineering. 62. 50-55. 10.17221/12/2015-RAE.

[13] N. Nowdehi, A. Lautenbach and T. Olovsson, "In-Vehicle CAN Message Authentication: An Evaluation Based on Industrial Criteria," 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, 2017, pp. 1-7, doi: 10.1109/VTCFall.2017.8288327.

[14] K. Kang, Y. Baek, S. Lee and S. H. Son, "Lightweight Authentication Method for Controller Area Network," 2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Daegu, 2016, pp. 101-101, doi: 10.1109/RTCSA.2016.58.

[15] R. Buttigieg, M. Farrugia and C. Meli, "Security issues in controller area networks in automobiles," 2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, 2017, pp. 93-98, doi: 10.1109/STA.2017.8314877.

[16] M. Bozdal, M. Samie and I. Jennions, "A Survey on CAN Bus Protocol: Attacks, Challenges, and Potential Solutions," 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, United Kingdom, 2018, pp. 201-205, doi: 10.1109/iCCECOME.2018.8658720.

[17] Hoppe, Tobias & Kiltz, Stefan & Dittmann, Jana. (2008). Security Threats to Automotive CAN Networks – Practical Examples and Selected Short-Term Countermeasures. Reliability Engineering & System Safety. 96. 11-25. 10.1016/j.ress.2010.06.026.

[18] Ueda, H. & Kurachi, Ryo & Takada, Hiroaki & Mizutani, T. & Inoue, M. & Horihata, S.. (2015). Security authentication system for in-vehicle network. 5-9.

[19] S. Sharaf and H. Mostafa, "A study of Authentication Encryption Algorithms (POET, Deoxys, AEZ, MORUS, ACORN, AEGIS, AES-GCM) For Automotive Security," 2018 30th International Conference on Microelectronics (ICM), Sousse, Tunisia, 2018, pp. 303-306, doi: 10.1109/ICM.2018.8704025.

[20] A. Vichare, T. Jose, J. Tiwari and U. Yadav, "Data security using authenticated encryption and decryption algorithm for Android phones," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, 2017, pp. 789-794, doi: 10.1109/CCAA.2017.8229903.

[21] K. Greene, D. Rodgers, H. Dykhuizen, K. McNeil, Q. Niyaz and K. A. Shamaileh, "Timestamp-based Defense Mechanism Against Replay Attack in Remote Keyless Entry Systems," 2020 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2020, pp. 1-4, doi: 10.1109/ICCE46568.2020.9043039.

[22] E. Wang, W. Xu, S. Sastry, S. Liu and K. Zeng, "Hardware Module-Based Message Authentication in Intra-vehicle Networks," 2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS), Pittsburgh, PA, 2017, pp. 207-216.

[23] Harel, A., Hezberg, A. Optimizing CAN bus security with in-place cryptography. In Proceedings of the SAE Connected and Automated Vehicle Conference Israel, Tel Aviv, Israel, 16–17 January 2019.

[24] Groza, B,Murvay, S.; van Herrewege, A.Verbauwhede, I. LiBrA-CAN: A lightweight broadcast authentication protocol for controller area networks. Lect. Notes Comput. Sci. Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform. 2012, 7712 LNCS, 185–200.

[25] P. Horster, M. Michels and H. Petersen, "Authenticated encryption schemes with low communication costs," in Electronics Letters, vol. 30, no. 15, pp. 1212-1213, 21 July 1994, doi: 10.1049/el:19940856.