IJCSN - International Journal of Computer Science and Network, Volume 7, Issue 1, February 2018
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

Rule-based Transceiver (RbT) Protocol for Security in Controller Area Networks

[1]Khaled Naga, [2]Ashraf Tammam, [3]Abdelmoneim Wahdan

[1] Department of Computer Engineering, AASTMT
Cairo, Egypt
eng.khalednaga@gmail.com

[2] Department of Computer Engineering, AASTMT
Cairo, Egypt
aft1972@yahoo.com

[3] Department of Computer Engineering, AASTMT
Cairo, Egypt
wahdan47@gmail.com

Abstract - Security researchers in Controller Area Networks (CAN) have addressed attacks targeting authentication, without being concerned about neither the protocol complexity nor the busload overhead. In this research, Rule-based CAN Transceiver (RbT) is introduced, proposing a security protocol targeting the validation of a set of rules, in addition to respecting real-time constraints in modern automotive CAN networks, and providing negligible busload overhead. In this research, modified CAN Transceiver hardware node is added to the network, acting as a network guard. Each node in the network will have to initiate a secure channel with RbT node, and during normal mode operation, the nodes will transmit a Message Authentication Code (MAC) as part of the data frame, with the MAC targeted to the RbT node. MAC will be built and validated based on a set of defined rules. RbT node will prevent the nodes from receiving the frame in case of invalid MAC by the transmission of CAN error frame, otherwise, it will allow normal frame reception by the network nodes. The protocol concept is proved and the busload overhead is implemented and proved using CANoe from Vector.

Keywords - *Automotive, Security, Controller Area Networks (CAN), Transceivers.*

1. Introduction

Thirty years ago, vehicles were pure mechanical systems that were fully operating through mechanical components. But around that time, the first software (SW) was introduced to vehicles [1] for controlling the ignition system of the vehicle and was a great evolution in the vehicle manufacturing industry. With the introduction of software to the vehicles industry, the process of mechanical parts replacement with electronics and software, a new era in vehicles industry started. 80% of innovations in automobiles are software-based [1]. With the introduction of Telematics and the connected cars to the internet, new features and bug fixes can be maintained remotely without physical contact with the car.

The Electronic Control Units (ECUs) hold sensors, actuators, electronic devices, and the processing micro-controllers for one or more systems. An example of a modern system that requires great processing in addition to the interaction with several standalone systems is the automatic parking system. The automatic parking system requires the control and communication with several ECUs including transmission, acceleration, braking, and steering ECUs. The number of ECUs in moderate class vehicles exceeds 70 ECUs exchanging more than 2500 signals [2] [3].

Several networks exist in modern vehicles, the most common network is the Controller Area Network (CAN). CAN has attractive characteristics that promoted it to be the most commonly used network. CAN operates as cooperative network, where the different connected nodes work independently while cooperating in sharing and broadcasting required data by other nodes. In addition to a relatively high data rate of 1Mbps, high noise immunity, and advanced mechanisms of error handling and fault confinement. But the CAN standard lacks security mechanisms.

No security is defined in CAN standard due to the fact that no external access to the CAN vehicle network is expected without physical access to the network. Currently, several wireless techniques in vehicles are introduced, including

connected vehicles to the internet through Telematics ECU, having the Telematics ECU on the network enables attackers to access the network. In addition to other wireless access methods, including connecting through Bluetooth wireless connection to the audio player ECU, and the wireless connection of Tire Pressure Monitoring System (TPMS) in modern vehicles. Where due to the introduction of wireless techniques in vehicles, the threat of attacking and accessing the vehicle's network is raised.

Several security techniques were introduced in literature, including software and hardware-based techniques. Significant busload overhead is added to implement the protocols especially to recover from attacks. In this research, a hardware-based protocol is introduced through a newly added network node as a modified CAN transceiver that validates the transmitted message using MAC, with the MAC is built based on a set of rules; the node is named as Rule-based CAN Transceiver node (RbT). For a node to operate on the network, it has to initiate a secure channel with RbT. Then for each transmitted message, a MAC is sent based on a set of rules defined in the research. RbT validates the MAC and transmits CAN error message in case of MAC invalidity, informing the receiver nodes to ignore the invalid message. Otherwise, it allows the message to be received.

The implementation and measurements of the introduced technique are done using Vector CANoe network development and simulation tool [4]. Also, Vector CANoe is used to measure the busload of the studied literature techniques.

The paper is organized as follows: Section 2 discusses a brief description of the CAN, security weakness in CAN, practical attacks on CAN, and previous related work. Section 3 presents the RbT protocol. Section 4 shows the experimental results and results' discussion. And, Section 5 concludes this paper.

2. Literature Review

This section introduces an overview of the Controller Area Network (CAN), the security weakness in CAN that causes security vulnerabilities. In addition to selected related work for security protocols in CAN networks.

2.1 Controller Area Networks (CAN)

CAN is an asynchronous serial bus network that connects devices, sensors, and actuators in a system or sub-system for control applications with high data rate reaching 1Mbps and payload data of 8-byte per frame [5]. CAN is considered as the most commonly used automotive serial protocol.

CAN is a Multi-Master hierarchy network, with the messages sent through broadcasting. Two modes of operation are available, standard mode with the message id as 11-bit allowing 2048 different messages on the bus, and extended mode with the message id as 29-bit allowing up to 635 million different messages. Also, several CAN frames are existing, data and remote frames for data transmission, error frame for error reporting, and overload frame for flow control. CAN also has high noise immunity feature as a result of using differential twisted-pair lines for data transmission.

Typical CAN applications includes Safety systems including Electronic Parking Brake, Body Control systems including Motor Control and Heating, Ventilation and Air Conditioning (HVAC), and Power Train systems including Electronic Throttle Control [5].

No security features were added to the CAN standard due to the fact that no attacks can be applied without a physical access to the network. But, this is no more valid with the addition of wireless interfaces in modern vehicles.

The conventional encryption techniques and digital signatures are not applicable to be used in CAN due to the limited payload data of 8-byte per frame and that the network nodes with CAN are considered as lightweight nodes with limited processing capabilities. So, MAC is a good candidate to be used for the addition of security features to CAN.

2.2 Security Weakness in CAN

Security weakness in CAN automotive networks arise from inherited weakness in CAN standard, and from deviations of security regulations.

CAN Standard Security Weakness. The absence of authentication mechanisms allows any node or attacker to send the desired messages without authentication. The broadcasting feature is a key factor in the strength of the CAN protocol, but it allows any node on the network to listen to the bus messages. The possibility of Denial of Service (DoS) attacks, due to arbitration feature of CAN. These reasons are raised from CAN standard features raising security weakness points that can be used in security attacks [6].

Security Regulations Deviations. ECUs re-flashing provides a possible scenario for the attacks, where the security keys used for re-flashing should not be the same

for all the ECUs in the network. Also, all the stored keys used for security purposes should not be easy to be acquired by external ECU. Keeping the network and ECU design respecting the security regulations can support in the protection against security attacks [7].

2.3 Practical Attack Attempts

Several attack attempts were performed by researchers using direct and indirect connections to the vehicle. In modern automotive era, an indirect connection is a possible way for security attacks.

Indirect access attacks were performed by compromising the vehicle wireless interfaces through the Bluetooth interface, cellular mobile network, and Tire Pressure Monitoring System (TPMS). Consequently, controlling the vehicle. The indirect attacks were analyzed and performed by Miller et al. [8] and demonstrated the attacks live [9] through compromising Uconnect 8.4AN/RA4 radio manufactured by Harman Kardon equipped in the attacked vehicle. Also Checkoway et al. [10] performed several attacks through a real road test.

Several direct access attacks were also performed through the direct connection with the CAN port in the vehicle. Koscher et al. [6], and Verdult et al. [11] [12] [13] were able to compromise the vehicle CAN network.

2.4 Related Work

This section presents the security protocols in CAN networks in literature with a brief description and protocol evaluation.

2.4.1 Time-based Techniques

Szilagyi et al. [14] introduced a voting technique, in addition to the transmission of a MAC byte per every existing node in the network. Combining the voting mechanism and the MAC per node, the protocol can be considered as a secure protocol. But the protocol requires timing mechanisms and consumes significant busload overhead.

Hartkopp et al. [15] introduced a time-based protocol "MaCAN", that adds a global timing to CAN networks through the use of Time Server hardware node in the network for time stamping, in addition to a Key Server hardware node for key distribution mechanism. The protocol uses 4-byte MAC for authentication, which is efficient but requires having 50% of the CAN message payload; adding significant message overhead and increasing the busload overhead. Also, the large length of the MAC might force the message to be sent over multi-frames. In addition, possible issues may occur during the synchronization messages' transmission.

2.4.2 Lightweight Software-based Techniques

Lin et al. [16] introduced an ID-Table based authentication protocol "Cyber-Security Protocol", where MAC is sent per actual receiver to the sent message, in addition to performing message stamping using counters. The protocol ensures message freshness using counters stamping and keeps the authentication security level with the use of MAC per receiver, but increase the busload significantly due to the MAC byte per receiver.

Hazem et al. [7] introduced a lightweight protocol "LCAP", that is built on the use of a 2 bytes "magic number". The protocol setup requires the use of significant number of CAN network IDs. Also, the re-synchronization of the nodes in case of parameters mismatch requires the exchange of several messages, affecting the busload and real-time constraints in modern automotive CAN networks.

Radu et al. [17] introduced a lightweight protocol "LeiA", that is built on sending a separate message for 8-byte MAC, and ensuring message freshness using counters. The protocol provides strong authentication protection, but requires significant busload overhead for sending the 8-byte MAC message and for the re-synchronization operation.

2.4.3 Hardware-based Techniques

Kurachi et al. [18] introduced a hardware-based solution for authentication "CaCAN", through the addition of a centralized monitoring node. The monitoring node is introduced to replace the existing CAN controller of the main gateway of the network, providing strong authentication protocol. The protocol does not add significant CAN IDs for the security implementation. The protocol is considered one of the earliest proposed hardware solutions in the literature. The main drawback is the vulnerability against Man-In-The-Middle (MITM) attacks due to the lack of sender authentication.

Research motivation is driven by the need to have a reliable security protocol in CAN networks while taking into account the limited capacity of CAN busload and respecting real-time constraints through a quickly performing protocol. RbT protocol is created to address the raised points, and described in Section 3.

3. RbT Protocol

RbT is a secure CAN transceiver hardware module introduced in the research as an added node to the existing CAN network; as a network guard. RbT is the core of the proposed RbT protocol providing security features that ensure messages authentication, authentication and authorization of the sender node, thus ensures message's freshness and sender node verification.

Each sender node in the network has to initiate a secure channel with RbT. After that, each sender node can transmit the data message as normal. In case of detection of vulnerability against authentication or authorization, RbT will cause the attacking message to be discarded by the network receiver nodes through the transmission of CAN error frame during the inter-frame space. Thus, the message will be discarded by the CAN transceivers of the receiver nodes before being processed by the node.

The protocol uses MAC for security, with the MAC holding encoded parameters for authentication and authorization. The protocol has a single MAC byte during the transmission of data messages, the MAC byte is transmitted in the payload data in case of CAN standard mode and in the extended ID field in case of CAN extended mode. Thus, the protocol adds negligible busload on the bus.

3.1 Protocol Parameters

To support and implement the security features, several parameters are required. The parameters are used for MAC generation and verification procedures.

The protocol parameters can be summarized as in table 1, with *n* is the network node and *m* is the transmitted message.

Table 1: RbT Protocol Parameters

Parameter	Length	Description
$PermKey_n$	64-bit	Permanent Encryption Key
$TempKey_n$	64-bit	Temporary Encryption Key
$SrcId_n$	16-bit	Source Node ID
RbT Id	16-bit	RbT Node ID
$SrcCnt_n$	64-bit	Source Node Counter
$MsgCnt_m$	64-bit	Message Counter
$SetupReqId_n$	11-bit	Setup Request CAN ID
$SetupRespId_n$	11-bit	Setup Response CAN ID
$MsgId_m$	11-bit	Message CAN ID

$SrcCnt_n$ is incremented for each successful message transmission from node *n*, while $MsgCnt_m$ is incremented for each successful transmission of message *m*.

User-defined parameters are additional optional parameters that are not mandated by the RbT protocol and needed by the network designer. The user-defined parameters are used as parameters for the MAC generation and verification.

3.2 MAC Generation Process

MAC in RbT protocol is not only used for message authentication, but also for sender authentication and authorization. This is achieved through the inclusion of the message and sender node parameters in the MAC.

The generation of MAC in RbT protocol is dependent on message type to be transmitted. Three message types are considered, the setup request message, the setup response message, and the normal data messages.

The generation parameters are provided to the intermediate MAC generator, then the intermediate MAC is encrypted using an encryption key, and finally, the MAC is truncated to 1-byte in case of data messages.

Intermediate MAC equation is illustrated in equation 1. The parameters are dependent on the message type.

$$\begin{aligned} &\textit{Intermediate MAC} = f(\\ &\quad SrcId_n \textit{ or RbTId, i.e. Transmitter Node ID} \\ &\quad SrcCnt_n, \\ &\quad MsgId_m \textit{ or SetupReqId or SetupRespId,} \\ &\quad MsgCnt_m, \\ &\quad TempKey_n\textit{, Only for Setup Response Message} \\ &\quad \textit{User Defined Parameters)} \end{aligned} \quad (1)$$

Encrypted MAC equation is illustrated in equation 2. $PermKey_n$ is used for setup mode messages, while $TempKey_n$ is used for data messages.

$$\begin{aligned} &\textit{Encrypted MAC} = f(\\ &\quad \textit{Intermediate MAC,} \\ &\quad PermKey_n \textit{ or } TempKey_n) \end{aligned} \quad (2)$$

3.3 MAC Verification Rules

The node that is responsible for doing the verification is the receiving node, which depends on the message type. RbT is the verification node in case of setup request messages and data messages, while the other network

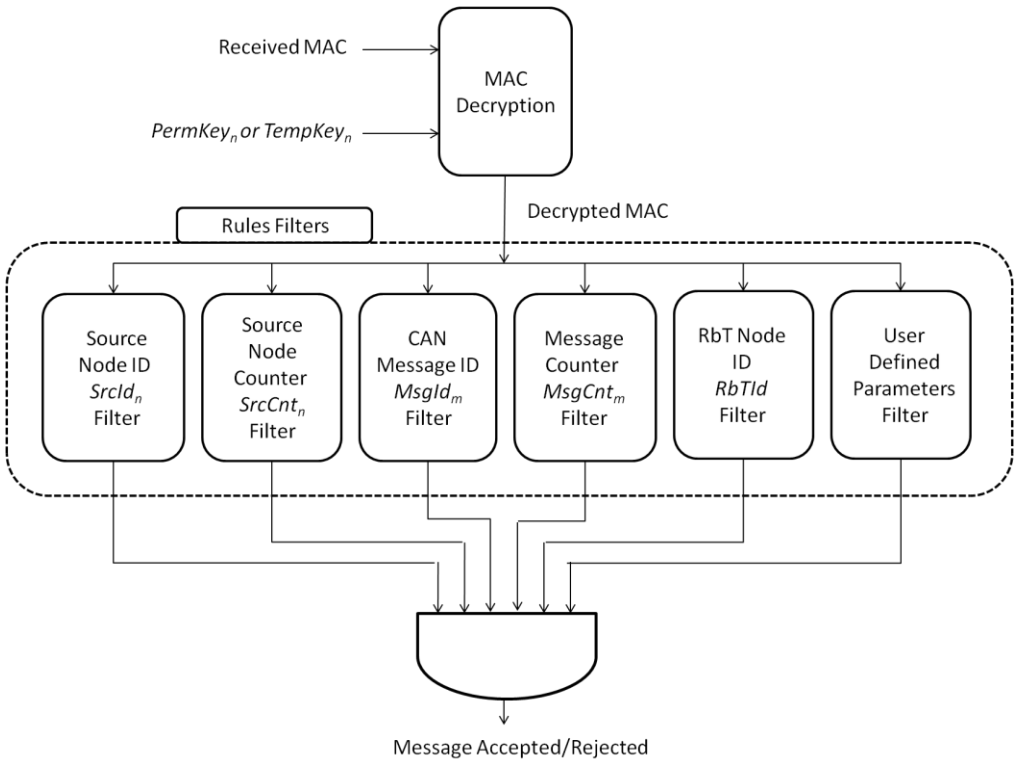

Fig. 1 RbT MAC Reception Representation

nodes are the receiver nodes for the setup response messages.

The receiving node should perform rules checking by applying rules filters on the received MAC. Since the received MAC holds information about the sender and the message, then the authorization of the sender and the authentication of the message and sender can be performed by the node.

RbT protocol performs rules checking and validation for the correctness and validity of the used encryption key, the transmitter node ID, the source node counter, the message ID, the message counter, and the user defined parameters if used. RbT MAC reception operation is represented is shown in figure 1.

3.4 Modes of Operation

RbT protocol has three modes of operation to satisfy the protocol features as the setup mode, the running mode, and the re-synchronization mode.

3.4.1 Setup Mode

During network nodes initialization, the nodes will need to set up a secure channel with RbT to be able to communicate the data messages on the bus. The secure channel setup is performed through a 2-way messages exchanged between the network node and RbT; Setup Request Message and Setup Response Message.

Setup Request Message is a CAN message with CAN ID $SetupReqId_n$ that is sent by the network node that needs to get access to communicate on the bus. The payload of the message is composed of 8-byte MAC. The MAC is so that RbT can authenticate and authorize the source node.

Setup Response Message is a CAN message with CAN ID $SetupRespId_n$ that is sent by RbT to the sender node as a response to set up the secure channel. The message holds an encrypted version of the temporary key $TempKey_n$ that will be used later by the sender node in the transmission of the data messages. The payload of the message is composed of 8-byte MAC. The MAC transmitted is used to enable the network node to authenticate RbT and to transfer $TempKey_n$ to the sender node from RbT.

The overall setup sequence is shown in table 2.

Table 2: RbT Setup Sequence

RbT Setup Sequence Procedure
At Sender Node Side - Transmit Setup Request Message:
1: Create Intermediate MAC
2: Create Encrypted MAC, using $PermKey_n$
3: Transmit the setup request message with CAN ID $SetupReqId_n$
At RbT Side - Receive Setup Request Message:
4: Authenticate the received message
5: Authenticate and authorize the sender node
At RbT Side - Transmit Setup Response Message:
6: Generate $TempKey_n$
7: Create Intermediate MAC, using $TempKey_n$ as an additional parameter
8: Create Encrypted MAC, using $PermKey_n$
9: Transmit the setup response message with CAN ID $SetupRespId_n$
10: Increment $SrcCnt_n$
At Sender Node Side - Receive Setup Response Message:
11: Authenticate the received message
12: Authenticate and authorize RbT
13: Extract $TempKey_n$
14: Increment $SrcCnt_n$

3.4.2 Running Mode

After a successful secure channel setup, the communication mode will turn into the running mode. Running mode is the mode where each sender node sends its data messages.

No additional messages are required to be sent when transmitting the data messages, only a single MAC byte is to be sent in the data message. The MAC is to be transmitted as a payload data byte in case of CAN standard mode and as part of the extended ID field of the CAN frame header in case of CAN extended mode.

Since CAN protocol is a broadcasting protocol, so the message is received by all the network nodes including RbT. RbT will read and process the MAC according to RbT protocol rules before the expiration of the inter-frame space.

In case of the validity of MAC according to the protocol rules, RbT will not transmit any additional messages on the CAN bus as a result of the received frame. So, the other network receiver nodes will read the message as normal.

In case of invalidity of MAC according to the protocol rules, then RbT will enter into re-synchronization mode.

3.4.3 Re-synchronization Mode

The protocol will enter re-synchronization mode in case of failure of the verification result. RbT will transmit a CAN error frame on the CAN bus during the inter-frame space. Thus preventing the network receiver nodes from processing the received data message, protecting the bus against possible attacks.

The overall Running/Re-synchronization Sequence is shown in table 3.

Table 3: RbT Setup Sequence

RbT Running/Re-synchronization Sequence Procedure
At Sender Node Side - Transmit Data Message:
1: Create Intermediate MAC
2: Create Encrypted MAC, using $TempKey_n$
3: Create Truncated MAC, using 1-byte truncation
4: Transmit the data message with CAN ID $MsgId_m$
At RbT and Receiver Nodes Side - Receive Data Message:
5: RbT and receiver nodes will receive the data message
At RbT Side - Process Data Message:
6: Authenticate the received message
7: Authenticate and authorize the sender node
8: If authentication and authorization passed
8.1: Increment $MsgCnt_m$ and $SrcCnt_n$
9: If authentication and/or authorization failed
9.1: Transmit CAN error frame on the bus
At Receive Nodes Side - After Inter-frame Space:
10: If no CAN error frame is transmitted by RbT
10.1: Start processing the data message
11: If CAN error frame is transmitted
11.1: The transceivers will automatically drop the data message

RbT re-synchronization mode diagram after invalid MAC detection is illustrated in figure 2.

3.5 RbT Protocol Characteristics

RbT Protocol Characteristics are described as follow:

Hardware Solution. RbT is an independent hardware chip in the network, and not a part of an existing network node.

Message and Sender Node Stamping. RbT uses counter-based stamping, due to the lack of timing in CAN. Two separate counters are introduced, the first counter is the source node counter that is incremented for every successful message transmission by the sender node, and the second counter is the message counter that is

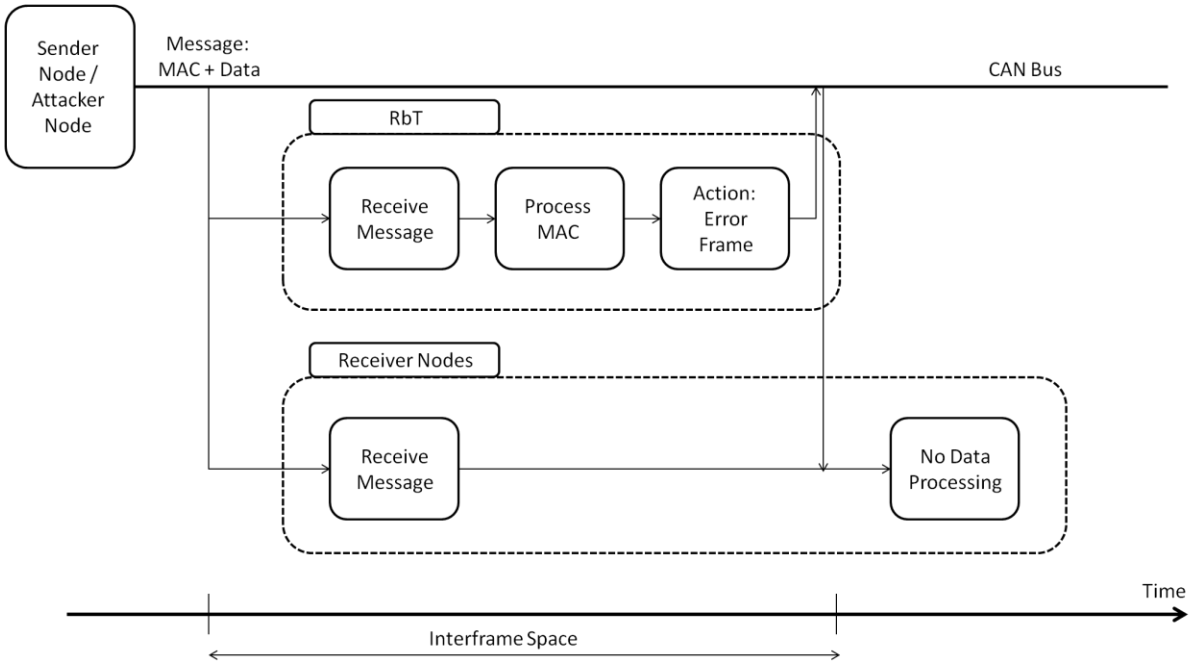

Fig. 2 RbT Running/Re-synchronization Mode - Error.

incremented per message for every successful message transmission. So, a counter is per node and a counter per message. And for security purposes, the counters are sent encoded as part of the MAC.

Fault Tolerance. RbT is not configurable through CAN, so it is immune to attacks by remote attackers unlike network ECUs which can be compromised by a remote attacker. And due to the authentication feature of the sender node, so RbT protocol is considered immune to MITM attacks.

Network Resources Required. Available network identifiers are considered as network resources, where for CAN standard, 11-bit standard ID and 29-bit extended ID versions exist. RbT can use either standard or extended protocols. RbT protocol requires minor additional CAN IDs. As only additional two CAN IDs per node for setup mode are required. With a typical number of nodes as 20 in typical networks, resulting in the addition of 40 CAN IDs. Which is minor compared to the available 2048 CAN IDs in standard CAN 2.0A and 536 million CAN IDs in extended CAN 2.0B.

Busload Overhead. In running mode, RbT protocol requires the addition of a single byte to the transmitted data messages, without the need to add additional messages for authentication. In re-synchronization mode, RbT protocol transmits the CAN error frame in the reserved inter-frame space where no data messages are allowed to be transmitted, adding no additional busload overhead. Thus, RbT protocol adds minor busload overhead.

Real-Time Response. In RbT protocol, the receiver nodes only receive the messages as normal as if no security protocol is implemented. Also, there is a low probability of the need for multi-frames due to the added MAC byte. So, RbT protocol is considered to respect real-time constraints.

In this research, RbT protocol is introduced and discussed. The protocol protects against attacks targeting authentication and authorization of the messages and the sender nodes, in addition to the protection against MITM attacks. The implementation results for RbT protocol to analyze the busload overhead is described in Section 4.

4. Results and Discussion

4.1 Implementation Platform

The implementation of the protocols is created, and the busload is measured using Vector CANoe 9.0.111 and CAPL scripting; with the busload measured using CANoe built-in functions. Testing cases are taken as of a high transmission rate of messages, with the messages are sent periodically. The CAN bus data rate configuration is used as CAN high-speed network with a planned data rate of 500kbps, and the test message as periodic messages with periodicity 1ms, 5ms, and 10ms.

4.2 Busload Measurements

The busload measurements are performed for software-based protocols, LeiA [17], Cyber-Security for CAN [16], and LCAP for both standard CAN mode and extended CAN mode [7], CaCAN hardware-based protocol [18], and RbT proposed hardware-based protocol for both CAN standard mode and CAN extended mode. The busload measurements are shown in table 4.

Table 4: Busload Measurements

Protocol	Busload - 1ms	Busload - 5ms	Busload - 10ms
No Security Protocol	13%	2%	1%
Literature Protocol's Measurements:			
Cyber-Security - No Attacks	16%	3%	1%
Cyber-Security - Attacks	66%	13%	6%
LCAP (Std Mode) - No Attacks	16%	3%	1%
LCAP (Std Mode) - Attacks	>100%	31%	15%
LCAP (Ext Mode) - No Attacks	13%	2%	1%
LCAP (Ext Mode) - Attacks	>100%	30%	15%
CaCAN - No Attacks	16%	3%	1%
CaCAN - Attacks	16%	3%	1%
LeiA - No Attacks	38%	7%	3%
LeiA - Attacks	>100%	27%	13%
Proposed Protocol's Measurements:			
RbT (Std Mode) - No Attacks	15%	3%	1%
RbT (Std Mode) - Attacks	15%	3%	1%
RbT (Ext Mode) - No Attacks	13%	2%	1%
RbT (Ext Mode) - Attacks	13%	2%	1%

It is expected that the measurements of the 5ms messages are exactly double that of the 10ms, and the measurements of 1ms messages are exactly 5 times that of 5ms, but a small deviation exists. The deviation exists due to a limitation in Vector CANoe's built-in function, as it provides only the integer part of the busload value.

4.3 Busload Analysis

The busload overhead is analyzed as shown in table 5. The busload results are normalized to the busload with no security protocol implemented and no attacks to get the overhead due to the protocol addition, the results are shown in figure 3. The normalized busload in case of an attack per each transmitted message is targeted to get the overhead of the protocol to recover from the attacks, and are shown in figure 4.

4.3.1 RbT Protocol Versus SW-based Protocols

The software-based protocols are commonly introduced in the literature due to the elimination of hardware units to implement the protocols. RbT protocol showed much better frame utilization through the use of a single MAC byte and much better busload overhead compared to the studied software-based protocols.

Table 5: Measurements Summary.

Protocol	SW /HW	Overhead Percentage - No Attacks	Overhead Percentage - Attacks
Literature Protocol's Measurements:			
Cyber-Security	SW	23%	407%
LCAP (Std Mode)	SW	23%	1092%
LCAP (Ext Mode)	SW	0%	1053%
CaCAN	HW	23%	23%
LeiA	SW	192%	938%
Proposed Protocol's Measurements:			
RbT (Std Mode)	HW	15%	15%
RbT (Ext Mode)	HW	0%	0%

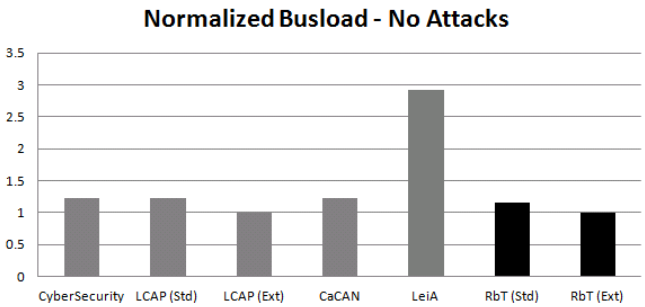

Fig. 3 Normalized Busload in case of No Attacks.

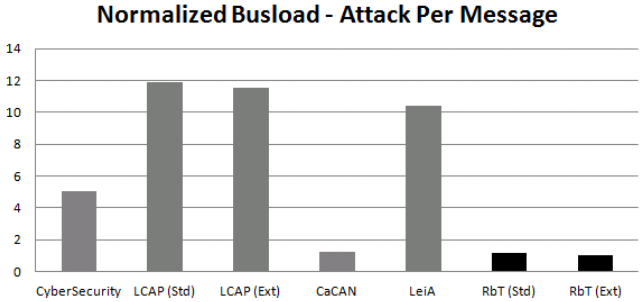

Fig. 4 Normalized Busload in case of an Attack Per Message.

Cyber-Security protocol frame utilization degrades as the number of message receivers increase. Also, the protocols' recovery mechanisms require the transmission of several re-synchronization messages thus increases the busload overhead in the bus.

LCAP protocol using CAN standard mode and CAN extended mode shows good frame utilization, where in CAN extended mode the MAC bytes are transmitted as part of the extended ID field. Thus the busload overhead in case of extended mode reaches nearly 0%. But, due to the large number of messages required for re-synchronization

in case of attacks or in case the nodes are out of synchronization, the busload overhead increases significantly. The huge busload overhead makes the protocol difficult to be used in practical networks.

LeiA protocol uses an 8-byte MAC. And, due to the transmission of the 8-byte MAC in a new CAN message, thus the busload overhead increases significantly reaching 192% in case of no attacks. In case of attacks, the busload overhead even increases dramatically.

RbT protocol showed much better frame utilization by consuming only a 1-byte for MAC in case of CAN standard mode and uses the extended ID field in case of CAN extended mode with no data overhead. The busload overhead in case of no attacks is found to be minimal of 15% compared to software-based protocols running in CAN standard mode, and the protocol keeps the same busload overhead in case of attacks due to the transmission of the CAN error frame during the inter-frame space. The busload overhead in case of running in CAN extended mode is found to be nearly 0% even in case of attack attempt per transmitted message. Thus RbT protocol shows much better results than the studied software-based protocols.

4.3.2 RbT Protocol Versus HW-based Protocols

CaCAN protocol uses 2-byte MAC, which decreases the frame utilization, as MAC consumed 25% of the payload. The busload overhead is shown to be around 23% in case of no attacks and in case of an attack per message. Thus CaCAN shows better results in frame utilization compared to software-based protocols. Also, CaCAN shows much better busload overhead results than the software-based protocols. But due to the lack of sender node authentication, CaCAN is vulnerable to MITM attacks.

RbT shows much better results in frame utilization compared to CaCAN, as it just uses a 1-byte for MAC in case of CAN standard mode, and uses the extended ID field in case of CAN extended mode, thus no payload bytes are used for MAC. This allowed better frame utilization than CaCAN. The busload is shown to be better than that of CaCAN, reaching only 15% for RbT compared to 23% for CaCAN, when using CAN standard mode. The busload overhead of RbT in CAN extended mode is the optimum, as it is nearly 0%. Thus, adding no overhead in case of CAN extended mode. So, allowing the network bus to run with full utilization as if no protocol is implemented. RbT performs sender node authentication, thus RbT protocol is considered to be invulnerable to MITM attacks.
From the obtained results, RbT protocol is shown to be of minor busload overhead in addition to the consumption of a single payload byte for MAC using CAN standard mode, or no payload consumption using CAN extended mode due to the use of the extended field for MAC transmission. Research conclusion and the intended future work is described in Section 5.

5. Conclusion and Future Work

RbT Protocol is a proposed hardware-based solution that depends on a new modified CAN transceiver. RbT MAC generation and verification rules are based on sender's parameters and the transmitted message parameters, ensuring protection against attacks targeting messages' authentication, authentication, and authorization of the sender nodes. RbT is proved to add negligible busload overhead while respecting strict real-time response constraints. RbT is shown to provide much better busload overhead results than the studied software-based protocols, and a better busload overhead results over the studied hardware-based protocol. In addition, RbT protocol is invulnerable against MITM attacks unlike CaCAN protocol due to the sender node authentication applied in RbT protocol.

Future Work. Several future works are planned, including the implementation of RbT protocol on real CAN networks to confirm CANoe measurements. In addition to addressing the practical challenges against RbT node configuration, including RbT node design and configuration challenges.

References

[1] M. Broy, I. H. Kruger, A. Pretschner, and C. Salzmann, Eds., Engineering Automotive Software, ser. 2, vol. 95. IEEE, 2 2007.
[2] A. Albert, Ed., Comparison of Event-Triggered and Time-Triggered Concepts with Regard to Distributed Control Systems, ser. 5. Proceedings of Embedded World, 2004.
[3] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, Eds., Trends in automotive communication systems, ser. 6, vol. 93. Proceedings of IEEE, 2005.
[4] Vector ELearning, "Controller area networks," 2016, https://elearning.vector.com/vl can introduction en.html.
[5] "In-vehicle networking," NXP, Tech. Rep., LIN/CAN/RF/FlexRay Technology.
[6] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, , and S. Savage, "Experimental security analysis of a modern automobile," in Security and Privacy (SP), 2010 IEEE Symposium, Oakland, CA, USA, 5 2010, p. 447462.
[7] A. Hazem and H. A. Fahmy, "Lcap - a lightweight can authentication protocol for securing in-vehicle networks," in 10th Int. Conf. on Embedded Security in Cars (ESCAR 2012), vol. 6, Berlin, Germany, 2012.

[8] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," 8 2015, http://illmatics.com/Remote
[9] C. Miller and C. Valasek, "Hackers remotely kill a jeep on the highway - with me in it," 2015, https://www.wired.com/2015/07/hackers-remotely-kill-jeephighway/.
[10] 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 20th USENIX Security Symposium (USENIX Security 2011). Berkeley, San Francisco, USA: Autosec, 8 2011.
[11] R. Verdult and F. D. Garcia, "Cryptanalysis of the megamos crypto automotive immobilizer," in USENIX Association, vol. 40, 2015.
[12] R. Verdult, F. D. Garcia, and B. Ege, "Dismantling megamos crypto: Wirelessly lock picking a vehicle immobilizer," in 22nd USENIX Security Symposium (USENIX Security 2013), 2013.
[13] R. Verdult, F. D. Garcia, and J. Balasch, "Gone in 360 seconds: Hijacking with hitag2," in 21st USENIX Security Symposium (USENIX Security 2012), 2012.
[14] C. Szilagyi and P. Koopman, "Low cost multicast authentication via validity voting in time-triggered embedded control networks," Workshop on Embedded System Security, 2010.
[15] O. Hartkopp, C. Reuber, and R. Schilling, "Macan - message authenticated can," in 10th Int. Conf. on Embedded Security in Cars (ESCAR 2012), vol. 6, Berlin, Germany, 2012.
[16] C. W. Lin and A. Sangiovanni-Vincentelli, "Cyber-security for the controller area network (can) communication protocol," in 2012 IEEE ASE International Conference on Cyber Security, Washington, DC, USA, 2012, p. 344350.
[17] A. Radu and F. Garcia, "Leia: A lightweight authentication protocol for can," vol. 9879, European Symposium on Research in Computer Security (ESORICS). Springer, 9 2016, pp. 283–300.
[18] R. Kurachi, Y. Matsubara, H. Takada, N. Adachi, Y. Miyashita, and S. Horihata, "Cacan - centralised authentication system in can," in 12th Int. Conf. on Embedded Security in Cars (ESCAR 2012), 2014.

Khaled NAGA received BSc degree in Electronics and Communication Engineering from Faculty of Engineering, Ain Shams University, Cairo, Egypt in 2008. He is a Software Architect and a Technical Consultant in Avelabs Egypt working for Automotive Tier-1 and Tier-2 companies including Delphi Automotive, Continental, Elektrobit, and Autoliv. He was a Principal Software Engineer in Valeo and IBM. He is also an experienced instructor conducting Embedded Systems specialized training in Delphi Automotive in the USA and several engineering institutes in Egypt including Information Technology Institute (ITI). His research interest includes embedded systems, real-time design, compilers, AUTOSAR, security, and automotive networks.

Ashraf TAMMAM received BSc degree in Computer Engineering from Military Technical College (MTC), Cairo, Egypt in 1994. And received his MSc and Ph.D. degrees in Computer and Systems Engineering from Faculty of Engineering, Ain Shams University, Cairo, Egypt in 2004 and 2011 respectively. He is an Assistant Professor of Computer Engineering at Arab Academy for Science, Technology & Maritime Transport (AASTMT), Cairo, Egypt. He was the Chairman of Information and Decision Support Center (IDSC), Egyptian Cabinet in 2014. His research interest includes computer networks, security, and cloud computing.

Abdelmoneim WAHDAN received BSc and MSc degrees in Computer Engineering from Faculty of Engineering, Ain Shams University, Cairo, Egypt in 1968 and 1972 respectively. And received his Ph.D. degree from École Centrale de Nantes, France in 1978. Since that, he worked as Assistant, Associate, and Full Professor of Systems and Computer Engineering in Faculty of Engineering, Ain Shams University, and on leave in KSU Saudi Arabia during 1985 to 1990. Professor Wahdan supervised many MSc and Ph.D. degrees in Egypt, Saudi Arabia, and France during his long career. And currently, he is with Computer Engineering department AASTMT, Cairo, Egypt. His research interest includes computer networks, computer hardware, embedded systems, automatic control, security, robotics, and other related areas.