# Improving the Security and Reliability of Embedded Networks With TTMAC-CAN

Ismail Ghodsollahee
Deppendable Distributed Embedded Systems (DDEms) Laboratory,
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran

Yasser Sedaghat
Deppendable Distributed Embedded Systems (DDEms) Laboratory,
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran

MohamadReza Pourmoghadam
Deppendable Distributed Embedded Systems (DDEms) Laboratory,
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran

*Abstract*—**The widespread use of distributed embedded systems to monitor and control large-scale industrial facilities (e.g., power plants, and electrical power grids), and the growing development of Internet of Things (IoT) have led to Industrial IoT (IIoT). In a typical IIoT, distributed embedded systems, composed of processors, sensors and actuators which are connected through embedded network protocols, are connected to the internet for remote management. However, embedded networks and therefore industrial facilities are very vulnerable to cyber-attacks. In this paper TTMAC-CAN, a novel scheduling algorithm is proposed which utilizes a multi-objective evolutionary algorithm to improve security, reliability, and performance of Time Triggered Controller Area Network (TTCAN) protocol. The proposed technique in comparison with TOUCAN reduces the risk of guessing the tag by about 86% and improve receiving cycle time by 98%.**

***Keywords—Message Authentication Code, Controller Area Network (CAN), Non-Dominated Sorting Genetic Algorithm (NSGA), Reliability, Security.***

## I. Introduction

The increasing use of embedded systems and the rapid growth of technology have led to the emergence of embedded networks. Embedded networks are fundamental infrastructures of industrial automation that made up of microcontrollers connection through communication protocols. Embedded systems have several limitations in the available memory, execution time, and power consumption that create challenges in implementation of security mechanisms at embedded networks[1]. Moreover, due to the remote management, these systems are connected to the internet, and hence securing the embedded network communication protocols is an essential requirement [2].

Controller Area Network (CAN) protocol is one of the mostly used embedded network communication protocols in Industrial Internet of Things (IIoT) [3]. This communication protocol allows the nodes to send and receive event-trigger messages with especial fuscous on real-time requirements. Although the CAN industrial bus is a differential serial bus and consists of twisted-pair wires which is noise resistant and fault tolerant, Electro Magnetic Interference (EMI) from the operational environment can cause transmission errors [4], and leads to violations in the real-time constraints of IIoT. Furthermore, CAN lacks authentication and protection against security attacks [5], and accordingly the attacker gains control of the application (e.g., a vehicle) through an external implementation or by physical contact through an On-Board Diagnostic (OBD) port in vehicles [6].

Various strategies have been proposed, to deal with these challenges. However, none of them consider improving throughput, security and reliability at the same time. CAN reliability is determined by a successful transmission of messages within their deadlines [7]. Several techniques have been proposed to improve the CAN reliability. For example, Shirai and Shimizu [8] present a FPGA-based technique to make the CAN bus resistant against EMI. Although a proposed technique in [8] improves reliability, their technique does not address network performance and security. In contrast, some papers, such as [9], [10], [11], [12], and [13], improve CAN network security whereas they do not consider performance or reliability improvement of CAN network.

Leen and Hefferman [14] propose Time-Triggered CAN (TTCAN) to provide deterministic response times and improve reliability of CAN in safety-critical systems. Although a series of papers [15-18] improves performance and reliability of TTCAN through scheduling, they do not consider security. As a result, to improve the authentication, reliability, and performance of the TTCAN network, in this paper TTMAC-CAN presented with focus on the message scheduling of TTCAN.

This paper seeks to optimize message scheduling of TTCAN protocol through NSGA-III-se evolutionary algorithm which proposed in [19]. The objective of this optimization problem is to improve reliability, security, and performance of TTCAN. In this scheduling, in order to improve reliability, temporal redundancy for real-time messages is considered, to improve security, session key for each two pair of nodes is provided, and moreover, to improve the performance, event-triggered messages gain maximized by minimizing the space occupied by time-triggered messages.

The remainder of this paper is organized as follows. In section II, security attacks and vulnerabilities in CAN network are introduced. Then the related studies are investigated in section III. Section IV evaluates the throughput and performance of different cryptography algorithms on 32-bit microcontrollers. Finally, Section V presents and evaluates the TTCAN_MAC by comparing the experimental results with the other related techniques.

978-1-7281-8566-8/20/$31.00 ©2020 IEEE

## II. Security Attacks in the CAN Network

The CAN protocol developed in the early 1980s by Robert Bosch GmbH for vehicular communication networks. Vehicle network were closed at the time of development of CAN, thereby this communication protocol lacks security requirements. Li, Wang, and Wu [20] demonstrate security vulnerabilities of CAN network. The security vulnerabilities described in [20] are shown in Figure 1.

The most important attack in CAN networks are suspension attack, fabrication attack, masquerade attack, and Denial of Service (DoS) attack. A suspension attack prevents a message from being transmitted (Figure 2(a)). In fabrication attack, unauthorized messages are injected into the network (Figure 2(b)). In the masquerade attack, the attacker releases a message with identifier of special node through another node (Figure 2(c)). In a DoS attack, a node releases a high-priority message to preventing other nodes from accessing the media (Figure 2(d)).

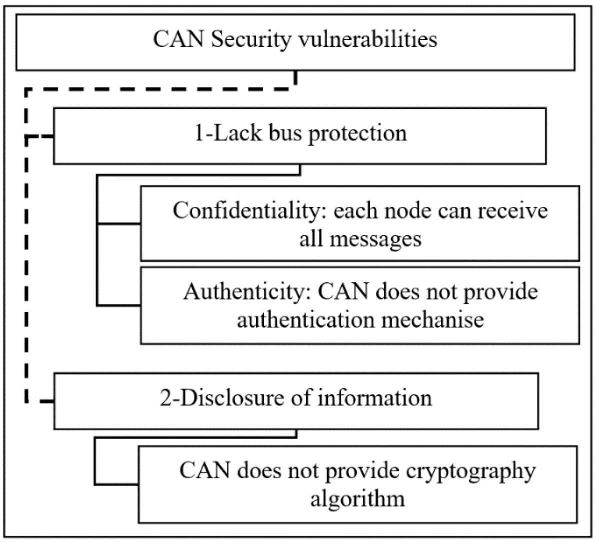
CAN Security vulnerabilities
1-Lack bus protection
Confidentiality: each node can receive all messages
Authenticity: CAN does not provide authentication mechanise
2-Disclosure of information
CAN does not provide cryptography algorithm

Fig. 1. Security vulnerabilities of CAN network

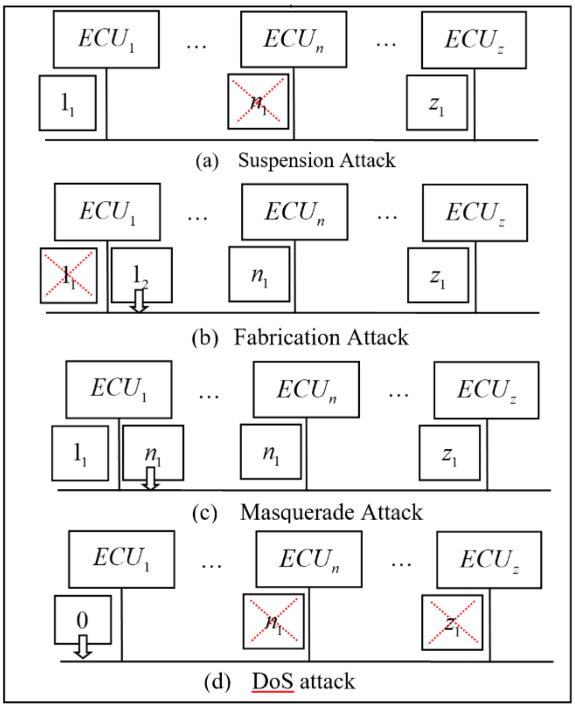

Fig. 2. Most important attack in CAN network

## III. Related Studies

Leen and Hefferman [14] have proposed TTCAN to guarantee successful transmission of all safety related messages. TTCAN is a higher layer protocol above standard CAN. In this extension of the standard CAN, the communication is based on the periodic transmission of a reference message by a time master [21]. In the period between two reference messages, there are several transmission time windows, and this distance between two reference messages, called a basic cycle. The transmission schedule of TTCAN stored in a System Matrix (SM), as shown in Figure 3. Rows of the System Matrix are called Basic Cycles and the columns are called Transmission Columns. Three types of time windows named exclusive, arbitration and free time windows, are defined in TTCAN. Exclusive time window is a time window for periodic messages. Arbitration time window is a time window for spontaneous messages, and free time window is a time window for further expansions [21].

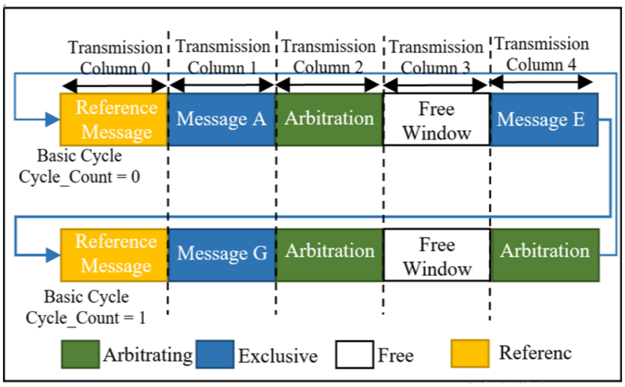

Fig. 3. System Matrix

A series of papers [15-17] have improved the performance of TTCAN, however they have not considered security or reliability improvement. In papers [15] and [16], the messages scheduling of the TTCAN has been optimized, with the Genetic Algorithm (GA). In these papers, objective of optimization is to make event-triggered messages gain the maximum by minimizing the space occupied by time-triggered messages. Moreover, Ti-Liang, Xiao-Bing, Xiao-Peng and Xian [17] have proposed real-time dynamic scheduling to decrease the network average delay and increase the network throughput.

In contrast, Xi Chen, Weijie Lv and Luyuan Liu [18] have improved reliability of TTCAN by optimizing the message scheduling via GA. In this paper, with considering the difference of reliability in three types of time window in TTCAN, the reliability has been improved by putting messages in different type windows.

As mentioned before, CAN lacks authentication and protection against security attacks. Several papers [9-13] have focused on improving security in CAN network. In [9-11], based on physical properties of CAN bus, several authentication methods have been presented. Pal-Stefan Murvay and Bogdan Groza, in [9], have presented a source identification technique based on voltage measurement of CAN bus. They use sampled voltages to identify a model for each node. In this technique, the bus is sampled, and these samples are passed through a low-pass filter. Then, this technique identifies a potential sender through convolution or Mean Square Error (MSE). In addition, without any knowledge about CAN node signals, existing nodes or security attack model, a voltage-based authentication technique for CAN with reinforcement learning have been proposed in [10].

Aside from techniques which analyse voltage properties of CAN bus, there are methods based on timing measurement. Bit-time-based intrusion detection [11] is an example of timing measurement techniques. Although authentication techniques based on physical properties have little computational overhead, these techniques employ hardware characteristics which is under the influence of environmental condition, and result in performance degradation of these methods.

Other studies, i.e., [12], [13], have proposed another class of authentication techniques based on Message Authentication Code (MAC). Yushev et al. in [12] have implemented Transport Layer Security (TLS) over CAN. Although TLS improves security of CAN network, it has high computation overhead, which makes this technique unsuitable for real-time applications. A technique, proposed in [13], offers a method called TOUCAN. In this method, the Chaskey MAC algorithm used for improving authentication in CAN networks. Chaskey is an efficient MAC Algorithm for 32bit microcontrollers which introduced in [22].

## IV. Implementation and comparison of cryptographic methods

In this section, four cryptographic algorithms including Data Encryption Standard (DES) [23], Triple DES (TDES) [24], Advanced Encryption Standard (AES) [25] and Chaskey MAC algorithm are evaluated on 32-bit ARM microcontrollers. As shown in Table I, evaluations are done on STM32F103C8T6 and STM32F030F4P6 from ARM Cortex-M3 and ARM Cortex-M0 series, respectively. These microcontrollers have resource limitations such as processing power and storage space.

TABLE I. Comparison of Cryptographic Methods

| | DES | TDES | AES | Chaskey |
|---|---|---|---|---|
| chosen Key Size (bits) | 56 | 56 | 256 | 128 |
| chosen Block Size (bits) | 64 | 64 | 128 | 128 |
| used ROM (Bytes) | 14883 | 15119 | 19232 | 11523 |
| used DRAM (Bytes) | 247 | 247 | 1384 | 269 |
| Computation Time (ms) STM32F030F4P6 | 40.43 | 130.29 | - | 0.328 |
| Computation Time (ms) STM32F103C8T6 | 20.10 | 60.42 | 30.4 | 0.126 |

Moreover, throughput of the mentioned cryptographic algorithms, i.e., DES, TDES, AES, and Chaskey MAC, on an ARM STM32F103C8T6 microcontroller has been shown in Figure 4. The throughput of these algorithms is calculated based on Equation 1, presented by [26]. In this equation, *Tp* is total plain text size in kilobytes and *Et* is the encryption time in seconds. As can be seen in the Figure 4, the throughput of the Chaskey MAC algorithm is higher than the other algorithms. Although, according to the Table I, DES and TDES encryption algorithms require little hardware resources, employing these algorithms result in more latency. Consequently, these algorithms are not suitable for real-time systems. Moreover, as seen in Table I, the ARM STM32F030F4P6 microcontroller does not have enough ROM (<16kbit) to implement AES. Furthermore, even though the ARM STM32F103C8T6 microcontroller has enough ROM to implement AES, this implementation has throughput about 450 Bytes/s. Given that the period of a control loop can be as low as 5msec [27], 450 Byte/s throughput is not enough for a common modern car, with approximately 70 CAN nodes [28]. Although, AES throughput can be improved by increasing the file size [26], the standard CAN message size is limited to a maximum of 8 bytes. Therefore employing AES in CAN communication network is not appropriate.

$$\text{Throughput} = \frac{\text{Tp (Kilobytes)}}{\text{Et (Seconds)}} \tag{1}$$

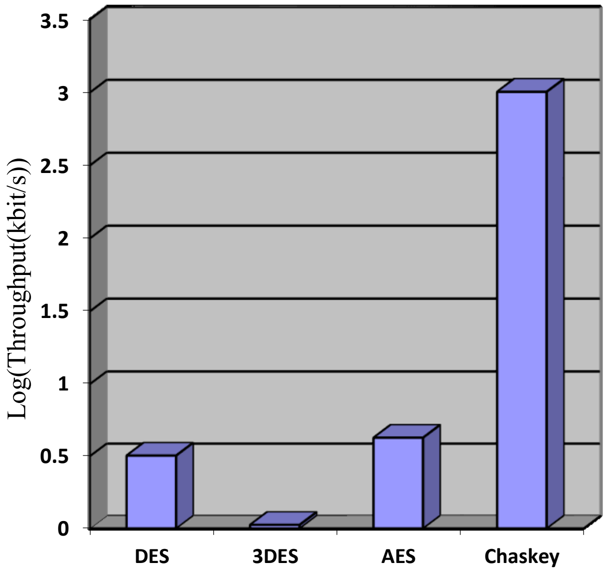

Fig. 4. Throughput of cryptographic algorithms

## V. TTMAC-CAN : The proposed technique

As mentioned earlier, CAN communication protocol, lacks security mechanisms. To resolve this issue, among the evaluated cryptographic algorithms in the previous section, Chaskey MAC algorithm has been selected. Although in [13], Chaskey MAC algorithm implemented over CAN, the deadline of real-time messages and the difference between MAC computation latencies in heterogeneous nodes have not been considered. As a result, in this paper, TTMAC-CAN with the ability to overcome the mentioned challenges, has been proposed. In this technique, different levels of security and real-time messages are defined for different levels of safety-critical functionality, including High-Security Hard-Real-Time (HSHRT) messages, Security or Real-Time Messages (SORT) and Normal messages. Examples of HSHRT messages are automated brake system messages in smart cars, which require authentication and should be transmitted within a specified deadline.

For HSHRT messages, an additional time window should be considered. Therefore, if these messages are corrupted, a retransmission is possible (as a temporal redundancy technique) which improves the reliability. In this case, 10 bytes are considered for Chaskey tag to maintain the security of this messages (Figure 5(a)). In contrast, SORT messages can be retransmitted if the retransmission overhead of these messages does not violate the deadline of other messages. Similar to TOUCAN technique [13], in TTMAC-CAN 24 bits are considered for Chaskey tag for these messages (as shown in Figure 5(b)).

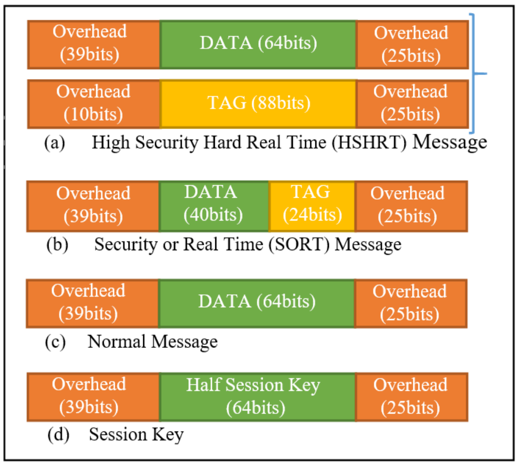

Fig. 5. Different type of messages in TTMAC-CAN

In TTMAC-CAN, System Matrix (SM) of TTCAN protocol, mentioned in [15], changes as shown in Figure 6. In the proposed technique, HSHRT, SORT and session key messages are transmitted in an exclusive time window and Normal messages are transmitted in the arbitration time window.

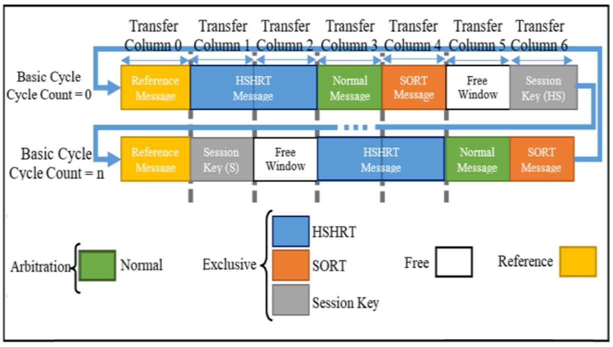

Fig. 6. Comparison of the output of different algorithm

In the first step of the proposed technique, initial information including CAN network baud rate, type (e.g. HSHRT and SORT) and deadline of messages should be provided for the scheduling algorithm. Then, the deadlines of messages should be recomputed, through Equation 2. This re-computation is due to the experimental study shows that if MAC computation latency in heterogeneous nodes is neglected, it leads to the violation of message deadlines.

$$D\_new_{m-n} = D\_old_{m-n} - (Lt_m + Lt_n + Lc_n) \quad (2)$$

In Equation 2, $D\_old_{m-n}$ is the deadline of message transmission from node $m$ to node $n$. $D\_new_{m-n}$ is new deadline of message transmission from node $m$ to node $n$. $Lt_m$ is the MAC computation latency of node $m$, $Lt_n$ is the MAC computation latency of node $n$ and $Lc_n$ is a delay caused by comparing computed tag with the received tag.

In the next step, several objectives, including security, reliability, and performance of TTCAN are optimized. So scheduling function are defined as Equation 3. This equation shows that each type of message scheduling will lead to different levels of performance, reliability, and security. The input of this function is the SM. In Equation 3, *Sec* indicates the level of security improvement which is determined by the number of available session keys in the schedule for each pair of nodes. *Per* indicates the performance which is determined by the ratio of the number of arbitration time window to the total number of time window, and *Rel* indicates the level of reliability improvement which is determined by the number of RT messages in the schedule which are able to use temporal redundancy.

$$\mathrm{Sec, Per, Rel} = \mathrm{ScheduleCo\ st}(\mathrm{MS}_1, \mathrm{MS}_2, ..., \mathrm{MS}_n) \quad (3)$$

$$\mathrm{MS}_i = [\mathrm{Type, Deadline, Dependenci\ es}] \quad (4)$$

In Equation 4, each message is defined by its type, deadline, and dependencies. In this optimization problem, objectives functions mathematically expressed as Equation 5. Nevertheless, before using the NSGA-III-se evolutionary algorithm mentioned in [19], we need to encode message scheduling as a chromosome (Figure 7).

$$\mathrm{maximize}\{\mathrm{S\ \ ec, Per, Rel}\} \quad (5)$$

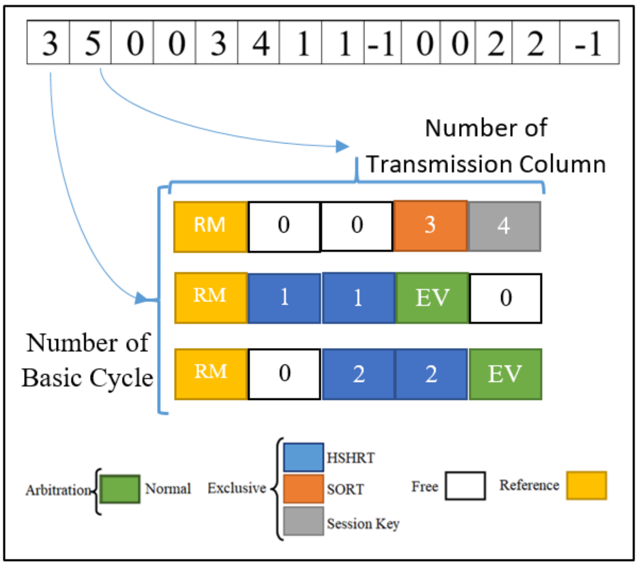

Fig. 7. Transforming scheduling to encoded chromosome

## VI. Implementation and Evaluation

In this section, the proposed technique evaluated theoretically and practically. Equations 6 and 7, introduced by [13], are employed to evaluate the security improvement of the proposed technique. In these equations, *tag_length* is the length of MAC-tag. As can be seen in the Figure 9, the proposed technique in comparison with TOUCAN reduces the risk of guessing the tag by about 86%. Guessing attacks are also performed to evaluate both techniques in such a way that after collecting messages, IDs, and tags, the attack program starts randomly generating tags, until the program realizes the success of attack by comparing the received and computed tag for a specific ID. The experiments show that after performing this attack for 13 hours and 18 minutes on the TOUCAN, used key in the Chaskey algorithm was founded. However, running 18 hours of the same attack on the TTMAC-CAN was unsuccessful.

$$\text{Risk of Guessing the Tag} = 2^{-\text{tag\_length}} \qquad (6)$$

$$\text{Limit Boundary Before Collision} = 2^{\frac{\text{tag\_length}}{2}} \qquad (7)$$

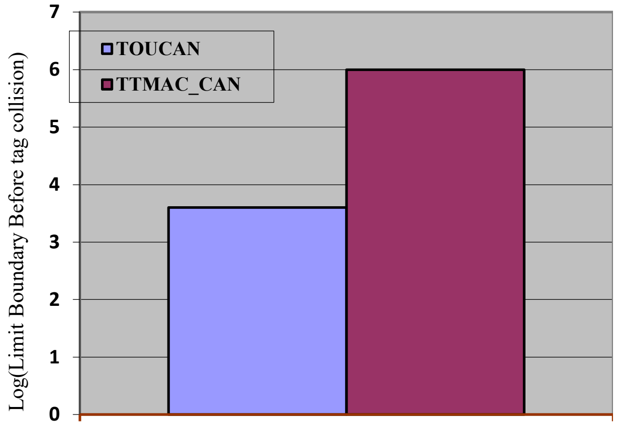

Fig. 8. Log (Limit Boundary before tag collision)

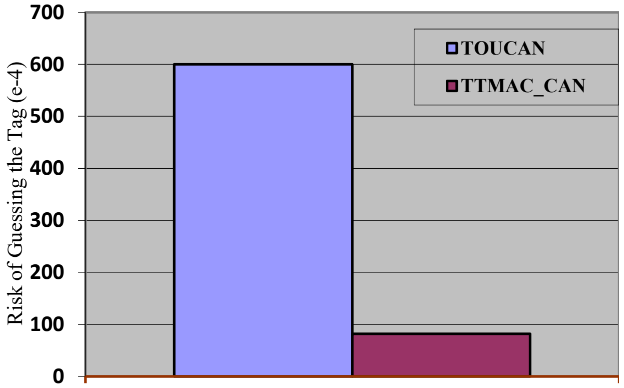

Fig. 9. Risk of Guessing the Tag (e-4)

In addition, SAE message set [29] is employed to evaluate the reliability of the proposed technique. The SAE message set is well known as a benchmark for CAN network. To do this evaluation, an embedded network was implemented as Figure 10. The employed SAE message set includes seven nodes (subsystems), however, duo to several limitations, e.g. hardware parts, in this implementation only four nodes were implemented. As shown in Figure 10, Node 1 composed of an STM32F407 microcontroller and an MCP2551 transceiver, which is responsible for sending vehicle controller (V/C) message set; Node 2 includes an STM32F205 microcontroller and an SN65HVD transceiver, which is responsible for sending driver inputs (Driver) message set; Node 3 composed of an STM32F103 microcontroller and an SN65HVD transceiver, which is responsible for sending brakes (Brakes) message set; and, finally, OBD-II node includes an OrangePi-2G IOT board, included an MCP2515 CAN Controller and an TJ1050 Transceiver, which is responsible for receiving all messages (similar to INS in the SAE benchmark) and report receiving cycle through Wi-Fi to a personal computer as a data logger.

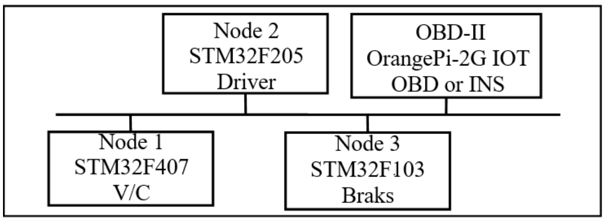

Fig. 10. Embedded network diagram

TABLE II. CONSTRAINTS AND DEPENDENCIES

| **Constraints** | | | | |
|---|---|---|---|---|
| message | ***Nodes*** | | ***Constraints*** | |
| | ***sender*** | ***receiver*** | ***Type*** | ***Deadline*** |
| M1 | N1 | N2, N3 | HSHRT | 10ms |
| M2 | N2 | N1 | SORT | 20ms |
| M3 | N2 | N4 | Normal | 30ms |
| M4 | N3 | N2 | HSHRT | 20ms |
| M5 | N2 | N3 | HSHRT | 20ms |
| M6 | N1 | N4 | Normal | 30ms |
| M7 | N1 | N2, N3 | Session Key | 500ms |
| M8 | N2 | N1 | Session Key | 500ms |
| M9 | N2 | N3 | Session Key | 500ms |
| **Dependencies** | | | | |
| ***messages*** | ***Dependency between messages*** | | | |
| M2 | M1, M4 | | | |
| M3 | M1, M4 | | | |
| M4 | M1 | | | |
| M4 | M1 | | | |
| **Number of Event base message in each Basic Cycle** | | | | |
| 1 | | | | |

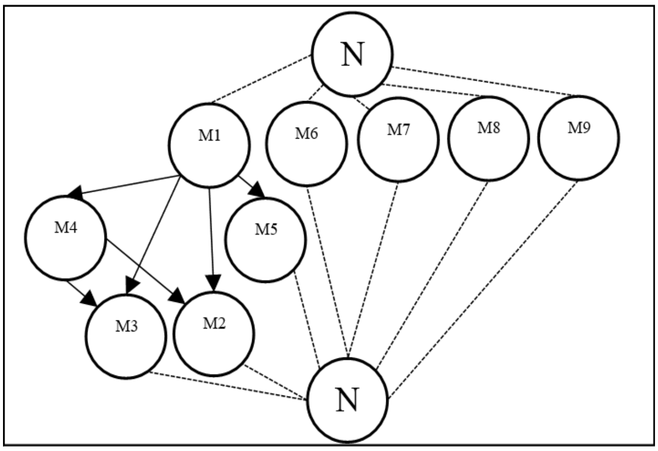

Fig. 11. Dependency graph of messages

Table II shows restricted set of SAE benchmark message set. Before scheduling, the baud rate of CAN network should be specified which is here considered 250 Kbit/s, and dependencies between messages should be obtained like Figure 11. It should be noted that if transmission of a message depends on another message, which was not transmitted or corrupted, it will reduce the overall performance of the system. Then message scheduling is optimized through NSGA-III-se evolutionary algorithm, which implemented with the Python programming language. The level of security, reliability, and performance improvement after 40 iterations shown in Figure 12. The optimal scheduling reported by NSGA-III-se evolutionary algorithm can be seen in the Figure 13. As shown in Figure 13, the possibility of temporal redundancy for messages and existence of a session key for each pair of nodes is determined through scheduling.

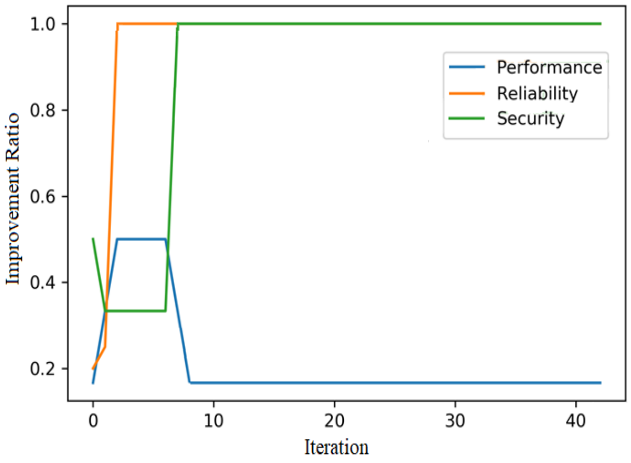

Fig. 12. Improvement of security and reliability through NSGA

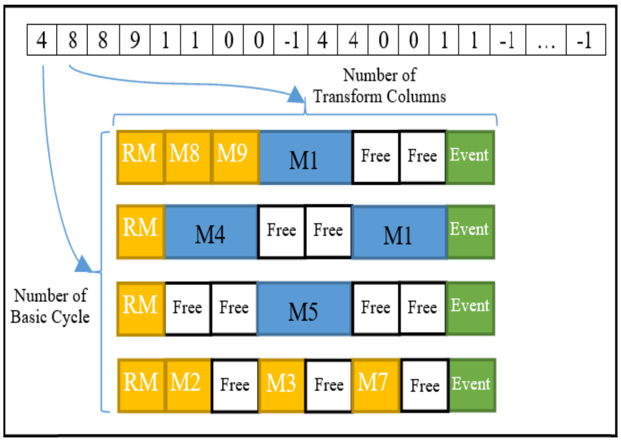

Fig. 13. Result of message scheduling with NSGA-III-se

After determining the number of Transfer Columns, the number of Basic Cycles, and the scheduling of messages, by applying NSGA-III-se evolutionary algorithm, hardware implementation is performed according to Figure 14. In the second part of the evaluation, the cycle of receiving hard real-time messages is measured to evaluate the number of deadline violations. It should be noted that the less deadline violations lead to better reliability. Figures 15, and 16 illustrate that the average receiving cycle time of TTMAC-CAN and TOUCAN techniques are 0.230ms and 8.320ms, respectively. The results show that the number of deadline violation in 80 cycles for TOUCAN is 24, whereas this number for the proposed technique is zero. Consequently, TTMAC-CAN in comparison with TOUCAN improves average receiving cycle time by about 98%. It should be noted that the negative values in Figure 16, are belong to messages which are received earlier than the specified time by the scheduler due to jitter.

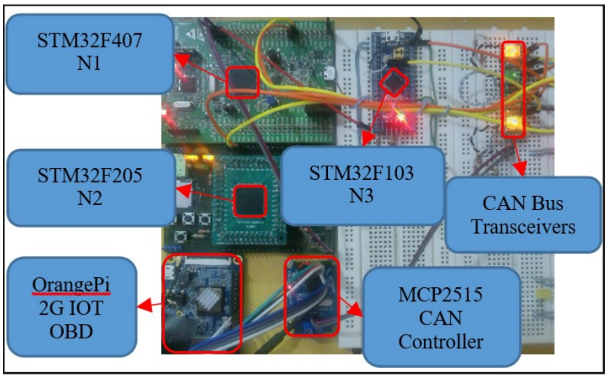

Fig. 14. Hardware Implementation

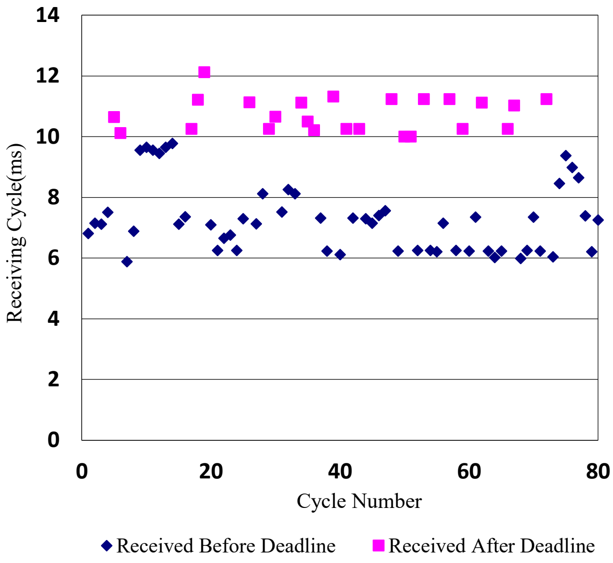

Fig. 15. Receiving cycle of M1 messages in TOUCAN

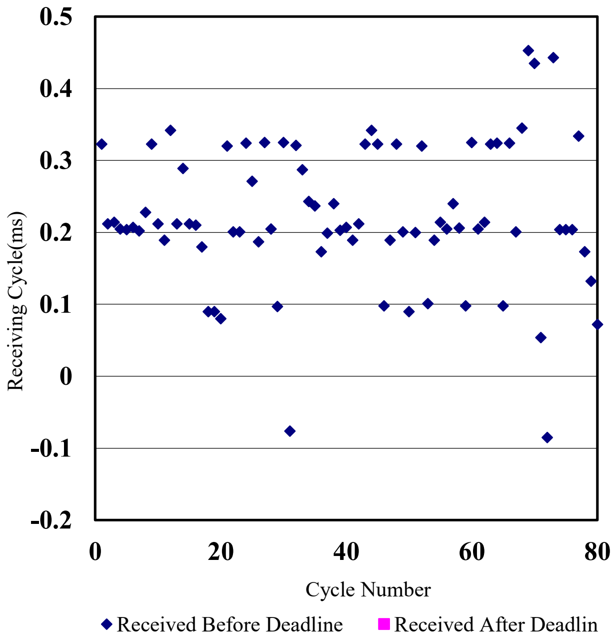

Fig. 16. Receiving cycle of M1 messages in TTMAC-CAN

## VII. Conclusion and Future Works

Due to the widespread use of embedded networks in the field of the IIoT and the security vulnerability of embedded networks, performance and security in IIoT systems has become a challenge. Moreover, due to the safety-critical nature of IIoT applications, improving reliability of embedded network is critical, too. Various strategies have been proposed, to improve reliability, security or performance of TTCAN. However, none of them consider improving throughput, security and reliability of TTCAN at the same time. In this paper, we have presented the TTMAC-CAN technique to resolve this issue. The proposed technique improves the security, reliability, and performance of embedded networks with TTMAC-CAN.

## References

[1] JiaYou Chen and Hong Guo, Wei Hu "Research on Improving Network Security of Embedded System," in *IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)*, Paris, France, 2019.

[2] L. M. R. Tarouco, L. M. Bertholdo, L. Z. Granville, L. M. R. Arbiza, F. Carbone, M. Marotta and J. J. C. d. Santanna, "Internet of Things in healthcare: Interoperatibility and security issues," in *2012 IEEE International Conference on Communications (ICC)*, Ottawa, ON, Canada, 29 November 2012

[3] Y. Kuang, "Communication between PLC and Arduino Based on Modbus Protocol," pp. 370–373.

[4] H. Aysan, R. Dobrin and S. Punnekkat, "Fault Tolerant Scheduling on Controller Area Network (CAN)," in *2010 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, Carmona, Seville, Spain, 07 June 2010.

[5] Z. Lu, Q. Wang, X. Chen, G. Qu, Y. Lyu and Z. Liu, "LEAP: A Lightweight Encryption and Authentication Protocol for In-Vehicle Communications," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, New Zealand, 28 November 2019

[6] Y. Zhang, P. Shi, C. Dong, Y. Liu, X. Shao and C. Ma, "Test and Evaluation System for Automotive Cybersecurity," in *2018 IEEE International Conference on Computational Science and Engineering (CSE)*, Bucharest, Romania, 27 December 2018.

[7] Shujun Yong, Lerong Qi,Yunhong Ma and Yifei Zhao, "Message Transmission Reliability Evaluation of CAN Based on DSPN," in *International Conference on Internet of Things as a Service Springer*, 2019.

[8] R. Shirai and T. Shimizu, "Failure Protection for Controller Area Network Against EMI Emitted by Buck Converter," in *2019 IEEE Applied Power Electronics Conference and Exposition (APEC)*, Anaheim, CA, USA, USA, 27 May 2019.

[9] Pal-Stefan Murvay and Bogdan Groza, "Source Identification Using Signal Characteristics in Controller Area Networks," *IEEE Signal Processing Letters,* vol. 21, no. 4, pp. 395-399, April 2014.

[10] Tangwei Xu, Xiaozhen Lu, Liang Xiao, Yuliang Tang and Huaiyu Dai, "Voltage Based Authentication for Controller Area Networks with Reinforcement Learning," in *IEEE International Conferences on Communications(ICC),* Shanghai, China, 2019.

[11] JIA ZHOU, PRACHI JOSHI ,HAIBO ZENG and RENFA LI, "BTMonitor: Bit-time-based Intrusion Detection and Attacker Identification in Controller Area Network," *ACM Transactions on Embedded Computing Systems*, vol. 18, no. 6, 2019.

[12] Artem Yushev, Mohammed Barghash, Minh Phuong Nguyen, Andreas Walz and Axel Sikora, "TLS-over-CAN: AN Experimental Study of Internet-Grade End-to-End Communication Security for CAN Networks," *IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd,* vol. 51, no. 6, pp. 96-101, 2018.

[13] Giampaolo Bella, Pietro Biondi ,Gianpiero Costantino and Ilaria Mateucci "TOUCAN A proTocol tO secUre Controller Area Network," in *Autosec 19: ACM Workshop on Automative Cybersecurity*, Texas, USA, 2019.

[14] G Leen and D. Hefferman, "TTCAN: a new time-triggered controller area network," *Elsevier Microprocessors and Microsystems*, vol. 26, no. 2, pp. 77-94, 2002.

[15] Shan Ding, Zhiqiang Xie and Xiaona Yin, "A GA-based Systematic Message Scheduling Method for Time-Triggered CAN," in *2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, China Shenyang, 2008. A

[16] Xin Qiao, Kun-feng Wang, Yuan Sun, Wu-Ling Huang, and Fei-Yue Wang, "A Genetic Algorithms Based Optimization for TTCAN," in *IEEE International Conference on Vehicular Electronic and Safety*, 2007.

[17] XIAO Ti-liang,LI Xiao-bing,T AN Xiao-peng and ZHOU Xian, "Real-time dynamic scheduling algorithm for TTCAN and it's realization," in *Real-time dynamic scheduling algorithm for TTCAN and it's realization*, Chengdu, China , 2010.

[18] Xi Chen, Weijie Lv and Luyuan Liu, "Optimization for the reliability of TTCAN bus based on Genetic Algorithms," in *2008 IEEE Conference on Cybernetics and Intelligent Systems*, Chengdu, China, 21-24 Sept. 2008.

[19] Zhihua Cui, Zhihua Cui, Yu Chang and Jiangjiang Zhang, Xingjuan Cai, Wensheng Zhang "Improved NSGA-III with selection and elimination operator," *Elsevier Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 23-33, 2019.

[20] Fang Li, Liafang Wang and Yan Wu, "Research on CAN Network Security Aspects and Intrusion Detection Design," in *SAE Technical*, 2017.

[21] Thomas Fuehrer, Bernd Mueller, Florian Hartwich and Robert Hugel, "Time Triggered CAN (TTCAN)," in *SAE Technical Paper, Robert Bosch GmbH*, 2001.

[22] Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel and Ingrid Verbauwhede, "Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers," *Selected Areas in Cryptography - SAC Springer*, vol. 8781, 2014.

[23] Simon J.Shepherd, "A high speed software implementation of the Data Encryption Standard", *Computers and security*, vol. 14,no. 4,pp. 349-357, 1995.

[24] D. Coppersmith, D. B. Johnson and S. M. Matyas, "A proposed mode for triple-DES encryption", in *IBM J. RES DEVELOP*,vol. 40, no. 2, MARCH 1996.

[25] V. Rijmen and J. Deamen, "Advanced encryption standard" in *Federal Information Processing Standard, Gaithersburg*, MD, USA, pp. 19-22, 2001

[26] Shaify Kansal and Meenaskshi mittal, "Performance evaluation of various symmetric encryption," in *IEEE International Conference on Parallel, Distributed and Grid Computing*, Solan, India, 2014.

[27] Koptez, H., "A solution to an Automative Control System Benchmark", Institut fur Technische Informatik, Technische Universitat Wien, research report 4/1994 (April 1994)

[28] Ralf Gümmer, Christopher Junk, George Rock, "A variant management based methodology for the requirements-engineering process of mechanical parts," in *Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering.* Springer, London, pp. 109–120, 2013

[29] K. Tindell and A. Burns, "Guaranteeing message latencies on control area network (CAN)," in *Proc. 1st Int. CAN Conf.,* Mainz, Germany,Sept. 1994, pp. 1–11.