Proceedings of the Fifth International Conference on Computing Methodologies and Communication (ICCMC 2021)
IEEE Xplore Part Number: CFP21K25-ART

Design and validation of CAN protocol with enhanced Security in electric vehicle

Deeksha. R
PG Scholar, Embedded Systems Technologies
Kumaraguru College of Technology
Coimbatore, Tamil Nadu, India.
E-mail: deeksha.19mes@kct.ac.in

Dr. Paramasivam. K
Professor, Electrical and Electronics Engineering Dept.
Kumaraguru College of Technology
Coimbatore, Tamil Nadu, India.
E-mail: paramasivam.k.ece@kct.ac.in

2021 5th International Conference on Computing Methodologies and Communication (ICCMC) | 978-1-6654-0360-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICCMC51019.2021.9418436

***Abstract*— Controller Area Network(CAN) which is specially designed for Automotive connectivity has a major drawback in the security of data transmission, where the data can be easily hacked. By hacking the data can be read, altered or even false data can be injected which will lead to a huge risk for the vehicle itself and the driver/passengers in the vehicle. To prevent this combining two algorithms in cryptography (AES and RSA) encryption to the messages in the CAN bus is provided. The data sent from the ECU will be encrypted using the AES algorithm, the key used by the AES will also be encrypted using the RSA algorithm. Decryption will be done by a reverse process where first the key of AES will be decrypted and then the data will be decrypted. By providing encryption and decryption the data can be transferred securely. This project also provides authentication which is achieved by sending acknowledgment signals. By providing encryption and authentication the Replay attack and Injection of forged frames can be avoided. Here the time taken for encryption, authentication, and decryption for different samples of the text of different sizes are also calculated. Results shows that proposed algorithm is more suitable for e-vehicle with better CAN protocol security.**

Keywords— CAN bus, Data transmission, Cryptographic, AES, RSA, ECU, Encryption, Decryption.

I. INTRODUCTION

As the name says, electric vehicles are those which operate on electric batteries. It is the only replacement for current generation automobiles to reduce environmental issues. They reduce emission of hazardous gases which plays a vital role in causing climatic changes and global warming. And with an increase in petrol price it serves as an best alternative where fueling is not a big issue and can be done through renewable resources. The other advantages of an electric vehicle is, it is connected digitally. Every connectivity is digitalized for convenience, space saving and for safety purpose. So for the digital connection, the CAN bus is used. The maintenance of electric vehicle is comparatively low. Performance and availability of storage options are better when compared to normal vehicle. It also provides advanced features for passengers convince. Electric vehicles do not cause noise pollution unlike normal vehicles.

II. CAN-BUS

It was first designed by Robert Bosh. It was developed to provide connectivity in electric vehicles between two or more controllers or other devices. It is a message-based protocol where the data are sent or received in the form of messages. So, by using CAN bus the space conception is reduced along with that the connection complexities are reduced in electric vehicles[1]. CAN simplifies the process, where through a single cable different ECU's could communicate. Few domains where CAN communication plays a major role are Power train, Chassis, Body, telemetric and Passive safety. Applications like power generation in engine and transmission through gear box, active safety, driving mechanism and assistance, body comfort, entertainment unit and safety mechanism requires CAN communication.

It is a multi master-multi slave protocol, because each and every ECU's connected to the CAN bus can act as both sender and receiver. To transfer data between ECUs it uses existing OSI reference model. When there are multiple messages to be transmitted, according to the priority of the messages the CAN bus transmits the data. Here, the communication is done through frames. A frame carries a sequence of bit or bytes of data. 10 bytes of messages is organized in a specific structure called frame. Based on identifier fields the CAN frame can be categorized as Standard frame and Extended frame. The Standard frame comprises of 11 bit identifier fields where as the Extended frame comprises of 29 bit identifier fields. The different types of message frames are Data frames, Remote frame, Error frame and Overload frame. The other advantages of CAN bus are it is flexible, reliable and low-cost. Even though there are many advantages in CAN bus it has its own limitations.

A. Drawbacks of CAN Bus

Drawbacks of CAN bus are

a. Broad caste
b. Authentication
c. Encryption

B. Types of Attack

Due to the above-mentioned drawbacks in CAN bus, the messages sent or received through the bus can be easily hacked. It is generally termed as an attack on the vehicle because the connectivity of the entire vehicle is done through the CAN bus which causes the risk to the entire vehicle. Attacks are classified into two categorized based on the way it is caused[1][2].

1. *Internal attack*
    a. Jamming
    b. Injection of forged frames
    c. Replaying original frames
    d. Denial of service attack
2. *External attack*
    a. Through software updates
    b. Through communication interfaces

III. LITERATURE SURVEY

Some of the related works regarding CAN protocols are, to secure the in-vehicular communication based on machine learning[3] where it is used to improve the support-vector of the machine model. Since this paper is to provide security for CAN bus in electric vehicles, a study on cyber attacks[4], hacking were acquiring the data's[5] and different hacking techniques[6] was done to have a better understanding. Another work, where two types of attacks were considered and the detection system for normal message and attack message was developed[7]. Securing CAN Bus involves networking where the messages are encrypted and decrypted so a study on various encryption and decryption algorithms was done[8]. Apart from electric vehicle combination of two algorithms was implemented in Bluetooth technology[9] and studies were done on different hybrid encryption algorithms[10]. A study on key distribution and ECU's[11] were necessary for implementing the AES and RSA algorithm. For data storage FOG computing[12] which tends to be a promising method was studied. [13]Provides methods for verifying the Integrity, where the replay messages can be avoided and Authenticity, where the messages might come from third party. For multi-frame CAN bus the AES algorithm provides encryption[14]. To identify injection attack an IDE based system is proposed which does not interrupt the CAN communication[15]. The other types of attacks like DoS, fuzzy and impersonation attacks is proposed in Deep learning model to avoid CAN traffic [16]. In Machine Learning a better understanding on the message passing through different ECU's is obtained[17]. The requirement of time in software[18] was explained to know the necessity of time.

IV. PROPOSED SYSTEM

In the proposed system, the data of the CAN bus is secured by encryption and decryption. The encryption and decryption are done by two algorithms, symmetric and asymmetric algorithms. The symmetric algorithm is AES and the asymmetric algorithm is RSA. Combining these two algorithms the data will be encrypted. Along with encryption authentication is also provided where the messages will be decrypted only after an acknowledgment signal is received from the sender ECU which is authorized. The time calculation of each process is calculated and execution time per word is also calculated.

The AES algorithm is opted because it is strongest and fastest. It also supports multi-frame communication. But the key used for encryption and decryption process is not secured, where it uses the same key for both the process. So to ensure security of the AES key the RSA algorithm is combined along with it. For RSA algorithm there is no need for securing its key because it has an separate key generation process.

A. AES Algorithm

Advanced encryption standard algorithm is an symmetric key algorithm which works on block cipher i.e., works on block of texts. It comprises of 128 bit data and has three different key sizes(128b, 192b and 256b). AES operates in an iterative manner and computes on byte wise operation where 128 bits of block will be equal to 16 bytes. Length of key will differ according to the number of iterative rounds taken place. Generally the AES algorithm will take place in 3 different rounds

128 bit key – 10 rounds
192 bit key – 12 rounds
256 bit key – 14 rounds

Each round uses different 128 bit round key, which is calculated from the original AES key. There are 4 sub-process under each rounds

1. Byte Substitution
2. Shift rows
3. Mix columns
4. Add round key

Encryption can be done in an orderly manner and decryption can be done in an reverse process.

B. RSA Algorithm

Rivest-Shamir-Adleman is an asymmetric cryptographic algorithm where 2 keys has to be generated. The two different keys are

1. Public key
2. Private key

RSA algorithm is also known as Public key cryptography. Here one key should be kept secret and the other key can be given to anyone. In RSA algorithm an separate key generation should be done for generating two keys which undergoes mathematical calculations. After key generation the public key will be in the sender end and the receiver end will be given with a private key.

C. Block Diagram

Figure 1 consists of a sender and receiver ECU which is denoted as ECU 1 and ECU 2. The plain text will be sent from ECU 1 which has to be encrypted and the ECU 2 will receive the plain text which will be decrypted. First, public and private keys will be generated using the RSA algorithm. The plain text will be encrypted using the key of the AES algorithm. Then the key of the AES algorithm will be encrypted using the public key of the RSA algorithm. In the receiver end after the decrypted message is received the ECU 2 will send an acknowledgment signal to ECU 1. Only after the verification is done the decryption process will take place if the verification process is not done the decrypted message will stay on hold and the decryption will not take place. After the authentication process gets complete. The encrypted key

along with the encrypted text will be transferred to the receiver end. Here first the AES key will be decrypted by the RSA private key and then the encrypted cipher text will be decrypted by the AES key. Finally ,the time for the encryption, authentication and, decryption process is calculated for different samples of the text of different sizes and the time difference is found.

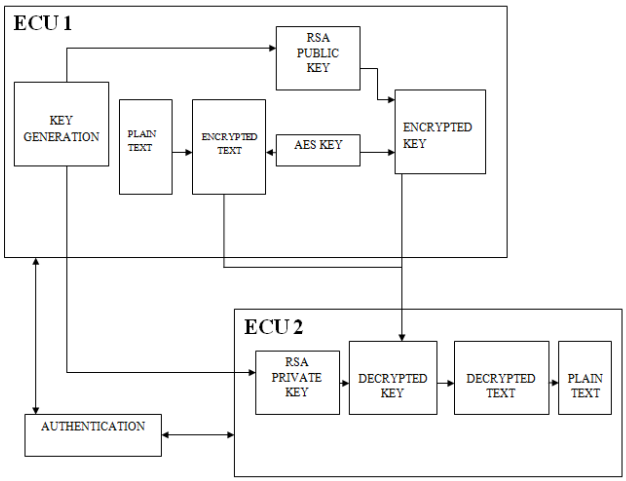

Fig. 1. Block diagram of proposed system .

V. OPERATING METHODOLOGY

The operation takes place in three steps, Encryption Authentication Decryption

A. Encryption

Encryption is a process of converting plain text into unreadable text or cipher text. By converting the normal message to cipher text the message cannot be read as such. This provides security for the message or the data sent through the CAN bus. Here, the plain text will be converted into cipher text using the AES key.

B. Decryption

Decryption is a process of converting the unreadable or the cipher text into readable or plain text. The cipher which is received through the CAN bus will only be processed only after converting the cipher text into plain text or else the ECU cannot process the message sent. Here, the cipher text will be covered into plain text using the AES key.

C. Authentication

Authentication is a process of providing security. Usually, the ECU's do not know whether the message obtained is received from the authorized ECU or not. So to provide or to overcome this the authentication is provided to the system. By giving Authentication the ECU will recognize the sender, if the sender is not authorized the message will not be decrypted.

D. Algorithm

STEP 1: Key Generation (public and private key) of RSA algorithm.
STEP 2: Encryption of plain text to cipher text using the AES key.
STEP 3: Encryption of AES key using RSA public key.
STEP 4: Transmission of both encrypted text and AES key to the receiver end.
STEP 5: Receiving the authentication signal
STEP 6: Decryption of AES key using the RSA private key
STEP 7: Decryption of plain text using the AES key.
STEP 8: Processing of plain text in application.

E. Implementation method

This method is implemented using Python language in Juypter Notebook which is an open source software. The libraries used are crypto, pycrypto, pycryptodomex, cryptography and fernet.

VI. RESULT

The result is obtained in the following manner.

a. First, the keys required will be generated "generating RSA public and private key and AES symmetric key" After obtaining the key, the message or the plain text which has to be encrypted will be asked.
b. Then the plain text will be encrypted using the AES key. The cipher text which is encrypted by the AES key will be displayed.
c. The AES key generated will be encrypted using the RSA public key.
d. The encrypted AES key will be displayed.
e. The authentication message will be displayed after the authentication process gets completed.
f. After the authentication message is obtained the AES symmetric key will be decrypted.
g. After decryption of the key, the original key will be displayed. Using the decrypted AES key the message will be decrypted and displayed.
h. Finally, the time taken will be displayed.

A. Output result

Figure 2 shows the simulation output of the proposed system. Where the word “embedded system” is encrypted and decrypted. The keys generated can also been seen along with the encrypted and decrypted text. The total execution time is also displayed.

```
Generate RSA public and Privite keys
Generate AES symmetric key
AES Symmetric Key:
d68757cea8eda22a6077ab814241b91f
Enter the message: embedded system
Encrypt the message with AES algorithm
AES cypher text:
b'n\xa5\ryO\x88z\xe2\x19L\xb2=}\x16\xec'
[1936, 11853, 23093, 21332, 13081, 21332, 25627, 16353, 1850, 23093, 16353, 1936, 1850, 10397, 10397, 1850, 11853, 11321, 2133
2, 21332, 1850, 14111, 23093, 12239, 1725, 10397, 1725, 12239, 14111, 29037, 12239, 3404]
Encrypt the AES symmetric key with RSA public key
Encryted AES symmetric key
[1936, 11853, 23093, 21332, 13081, 21332, 25627, 16353, 1850, 23093, 16353, 1936, 1850, 10397, 10397, 1850, 11853, 11321, 2133
2, 21332, 1850, 14111, 23093, 12239, 1725, 10397, 1725, 12239, 14111, 29037, 12239, 3404]
Decrypt the AES Symmetric Key using RSA private
AES Symmetric Key:
d68757cea8eda22a6077ab814241b91f
Decrypt the message using the AES symmetric key
Decrypted message:
embedded system
Press ENTER to exit
Execution time is 8.527342557907104
```

Fig. 2. Output result for simple plain text

B. *Authentication result*

Figure 3 shows the result of authentication where an acknowledgment signal "true" is obtained. Array is used in fernet library for generating the acknoledgment signal.

```
0.88453657 0.04549967 0.86916613 0.79906016 0.96639517 0.07861806
0.05525862 0.65228445 0.86166275 0.4564726  0.32010824 0.28803068
0.40244254 0.15003747 0.22021937 0.84224287 0.57787094 0.1234712
0.55142495 0.60390173 0.60969308 0.46916769 0.41069203 0.86489865
0.31371884 0.94003359 0.80896399 0.70330967 0.0697057  0.47493907
0.21215985 0.63756842 0.76986741 0.14641326 0.14136737 0.60482712
0.24440766 0.67937768 0.78386619 0.34150024 0.71675363 0.08768676
0.58384974 0.62098521 0.85196959 0.30140248 0.75412549 0.39736031
0.22290674 0.81893334 0.30341331 0.24336791 0.77597907 0.12750002
0.57030535 0.27932451 0.49727847 0.60066223 0.46510219 0.68028918
0.65861606 0.32461887 0.39475437 0.52967696 0.64431606 0.77727836
0.46101944 0.50425614 0.4620854  0.72194474 0.53701896 0.41785274
0.7333368  0.87997463 0.46250322 0.72189854 0.91126855 0.61828937
0.25245475 0.76894382 0.47581266 0.7245129  0.45392925 0.37543801
0.34546469 0.00421063 0.12456797 0.12217599 0.38616679 0.49467031
0.22529389 0.86046145 0.35316702 0.39273309 0.7983596  0.88840928
0.26613665 0.92400152 0.75793343 0.52753271 0.68893336 0.21518226
0.21162544 0.25811759 0.52709198 0.97886771 0.56191224 0.93513898
0.15382314 0.5643352  0.368532   0.04968147 0.19229298 0.32313435
0.01859461 0.34307001 0.78398998 0.94600177 0.18526688 0.79398213
0.83350667 0.54740175 0.25993343 0.65467756 0.02685772 0.49965691
0.70423212 0.07549614 0.60193771 0.83636806 0.618219   0.24569299
0.65051132 0.84119782 0.85734507 0.01382563 0.37900405 0.04293067
0.99736888 0.22184198 0.05025061 0.90185116 0.04338083 0.49278723
0.54968397 0.58131071 0.11315337 0.61555006 0.15254794 0.12070899
0.64155538 0.76394825 0.06949265 0.0117922  0.31829298 0.04038611
0.96379496 0.64066355]
True
--- 0.557553768157959 seconds ---
```

Fig. 3. Authentication Result

C. *Execution Time calculation*

Execution time for various process such as encryption, decryption and authentication are calculated for different set of words and listed in the Table I. Total time taken for entire process is also found. Table I shows the results obtained for time calculation. Authentication takes constant time for any number of words while encryption and decryption time increases when number of words increased.

Table I. Execution Time calculation in seconds

No. of words	Encryption time(s)	Authentication time(s)	Decryption time(s)	Total time(s)
100	10.96	0.55	7.450	18.97
500	18.60	0.55	16.15	35.30
1000	30.05	0.55	28.48	59.09
2000	43.59	0.55	40.29	83.87
5000	55.98	0.55	52.74	108.73
10000	90.71	0.55	87.46	178.18
20000	109.63	0.55	105.79	215.42
50000	123.58	0.55	120.78	244.36
100000	209.04	0.55	205.51	414.55

Figure 4 shows execution time versus words in graphical representation. It shows that encryption and decryption tile are increased while authentication time is constant.

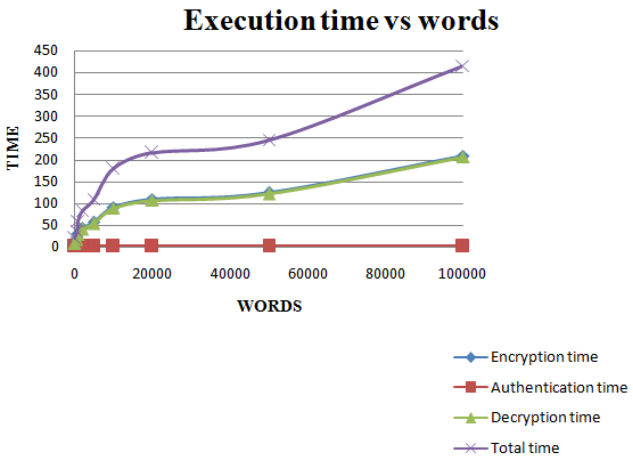

Fig. 4. Time(sec) vs Word graph

In table II, time required to process one word is calculated to show the effectiveness of algorithm. It shows that time per word is rediced with respect to increase in words. More words are required to process in practical case and the executin time should not disturb the performance of vehicle while in movement in faster speed. As veicle gets ignited, more amount of data are interchanged for processing and decesion making. Hence the reduction in time-per-word is adavantageous for practical situation in e-vehicle with better secured CAN protocol.

Table II. Execution Time per word calculation

No. of words	Encryption time(ms)/word	Decryption time(ms)/word	Total time(ms)/word
100	109	74	190
500	37	32	71
1000	30	28	59
2000	22	20	41
5000	11	10	22
10000	9	9	18
20000	5	5	8
50000	2	2	5
100000	2	2	4

Figure 5 shows the graph representation for time-per-word calculation of encryption and decryption process.

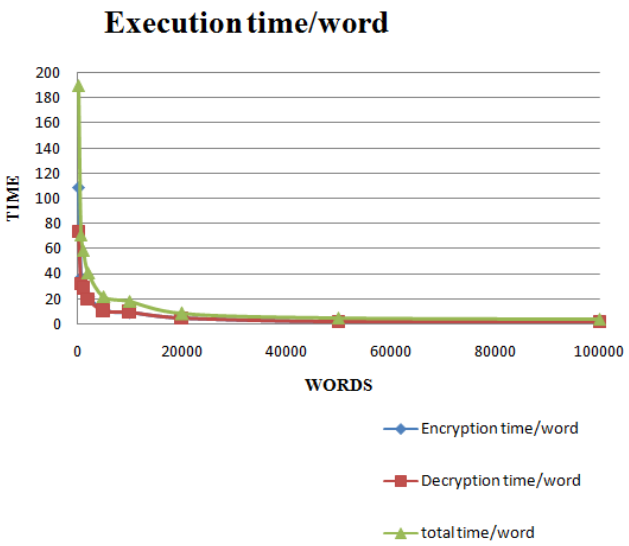

Fig. 5. Time taken per word (ms)

VII. CONCLUSION

In this project, combining two algorithms(AES and RSA) the encryption and decryption of messages in the CAN bus is ensured. Authentication is also provided in this project which is an important and major drawback of the CAN bus. By providing encryption and authentication the Replay attack and Injection of Forged frames can be avoided which has been simulated in this project. Along with this calculation of the encryption time, authentication time, and decryption time is done separately for various samples of the text of different sizes and time taken for each word to process is calculated, where when the number of words increases the time for each word decreases in the process. This shows that the proposed CAN protocol security algorithm is more suitable for e-vehicle with voluminous data processing.

VIII. FUTURE WORK

Here, only the implementation of the AES and RSA algorithm was done. In future an comparative analysis could be given with other algorithms or it could be implemented with other advanced algorithms. Proposed algorithm is to be implemented in simulation environment for better validation.

IX. REFERENCE

[1] Basker Palaniswamy; Seyit Camtepe; Ernest Foo; Josef Pieprzyk, "An Efficient Authentication Scheme for Intra-Vehicular Controller Area Network", IEEE Transactions on Information Forensics and Security (Volume: 15), March 2020.

[2]Mohammad Raashid Ansari; Shucheng Yu; Qiaoyan Yu, "IntelliCAN: Attackresilient Controller Area Network (CAN) for secure automobiles", 2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), Oct. 2015.

[3] Mamdooh Al-Saud; Ali M. Eltamaly; Mohamed A. Mohamed; Abdollah KavousiFard, "An Intelligent Data-Driven Model to Secure Intra-vehicle Communications Based on Machine Learning", IEEE Transactions on Industrial Electronics(Volume: 67, Issue: 6, June 2020), July 2019.

[4]P. Darby and R. Gottumukkala, "Decentralized Computing Techniques in Support of Cyber-Physical Security for Electric and Autonomous Vehicles," 2019 IEEE Green Technologies Conference(GreenTech), Lafayette, LA, USA, 2019, pp. 1-5, doi: 10.1109/GreenTech.2019.8767163.

[5] Prajakta Pimple, "Sniffing the Automotive CAN Bus for Real-time Data-logging and Real-Time Diagnostics Display",2018 International Conference on Smart Electric Drives and Power System (ICSEDPS), June 2018.

[6] Sam Abbott-McCune; Lisa A. Shay, "Techniques in hacking and simulating a modern automotive controller area network", 2016 IEEE International Carnahan Conference on Security Technology (ICCST), Oct. 2016.

[7] Iqra Nazakat; Khurram Khurshid, "Intrusion Detection System for In-Vehicular Communication", 2019 15th International Conference on Emerging Technologies (ICET), Dec. 2019.

[8] 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.

[9] Marwan Ali Albahar; Olayemi Olawumi; Keijo Haataja; Pekka Toivanen, "Novel Hybrid Encryption Algorithm Based on Aes, RSA, and Twofish for Bluetooth Encryption," Journal_of_Information_Security, April 2018.

[10] E. Jintcharadze and M. Iavich, "Hybrid Implementation of Twofish, AES, ElGamal, and RSA Cryptosystems," 2020 IEEE East-West Design & Test Symposium (EWDTS), Varna, Bulgaria, 2020.

[11]I. E. C. Roca, J. Wang, J. Du and S. Wei, "A Semi-centralized Security Framework for In-Vehicle Networks," 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 2020, pp. 1-6, doi: 10.1109/IWCMC48107.2020.9148360.

[12]Bhalaji,N. “Fog Computing-A Rasperry Pi Decentralized Network.” Journal of Information Technology 2, no. 01(2020): 27-42.

[13]T. Dee and A. Tyagi, "Secure CAN for Connected Vehicles," 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2020, pp. 1-6, doi: 10.1109/WF-IoT48130.2020.9221400.

[14]X. Yao, H. Sun, X. Jin and Y. Ding, "An Optimization Method for Encrypting CAN Messages," 2020 39th Chinese Control Conference (CCC), Shenyang, China, 2020, pp. 7643-7648, doi: 10.23919/CCC50068.2020.9189356.

[15]Q. Wang, Z. Lu and G. Qu, "An Entropy Analysis Based Intrusion Detection System for Controller Area Network in Vehicles," 2018 31st IEEE International System-on-Chip Conference (SOCC), Arlington, VA, USA, 2018, pp. 90-95, doi: 10.1109/SOCC.2018.8618564.

[16]Y. Lin, C. Chen, F. Xiao, O. Avatefipour, K. Alsubhi and A. Yunianta, "An Evolutionary Deep Learning Anomaly Detection Framework for In-Vehicle Networks - CAN Bus," in IEEE Transactions on Industry Applications, doi: 10.1109/TIA.2020.3009906.

[17]U. Ezeobi, H. Olufowobi, C. Young, J. Zambreno and G. Bloom, "Reverse Engineering Controller Area Network Messages using Unsupervised Machine Learning," in IEEE Consumer Electronics Magazine, doi: 10.1109/MCE.2020.3023538.

[18]Iddalagi, Pavan. “SDWAN-Its Impact and The Need of Time.” Journal of Ubiquitous Computing and Communication Tecnlogies (UCCT) 2, no. 04 (2020):197-202.