Volume: 8 Issue: VII Month of publication: July 2020

DOI: http://doi.org/10.22214/ijraset.2020.7021

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

CAN Encryption for Automotive Application

Mr. Kiran Ashok Bachhav[1], Miss. Tejal Ashok Jadhav[2]
[1, 2]Software Engineering and Systems, John Deere. India. Pvt. Ltd Pune, India

Abstract: The paper presents an overview of strong security to the CAN communication network for Embedded Application using AES encryption.

Index Terms: CAN Communication, Automotive Network, CAN Encryption, Automotive Network Encryption, AES Encryption.

I. INTRODUCTION

Control Area Network (CAN) is a robust communication protocol that helps in communicating information across various microcontroller boards (ECUs) in Automotive vehicles. Automated guided vehicles reduce the operator's work intensity, resulting in enhanced efficiency and increased operator safety. Besides engine the ECUs are used in various functional areas such as transmission, hydraulic system, and displays. To effectively control the various systems, it is necessary to ensure the communication between the control units and this communication done via CAN bus because of its robustness. But CAN is a low-level protocol and does not support any security feature.

CAN networks are Private network. The network has been embedded in machinery without much connection to other outside networks or the Internet. In this case, a physical attack by hackers is only possible – a hacker would need physical access to the machine to get access to the CAN subsystem.

However, nowadays the CAN subsystem is no longer independent or enclosed. Most frequently, bridges and gateways to other networking technologies are added, including connections to devices that have access to the Internet. Some examples of these devices are remote access devices for maintenance or diagnostics and multimedia servers like those in use in some automotive applications.

When we add such types of devices that will give a gateway to the Internet or offer wireless communication options like Bluetooth and Wi-Fi, this will increase the possible way of attacks to the CAN network.

CAN is also a popular communication channel for software updates so there is a possibility of external intrusion. Also the data available can be used for a purpose which Automotive manufacture may not want other entities to use that data.

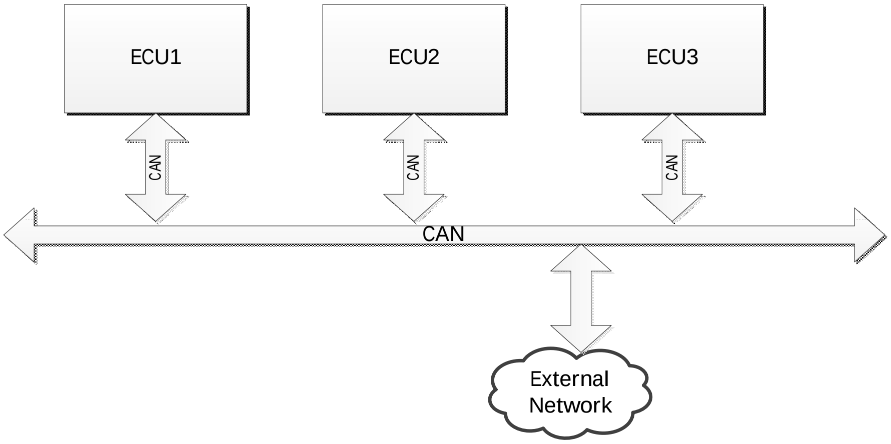


Figure 1 - Basic ECU Connection

II. SYSTEM DESCRIPTION

Certain embedded applications produce analytical raw data as an output for customers. The Automotive embedded systems work on CAN to communicate data between ECUs as well as the external world. The PC based application or the Online Algorithm reads these messages and reproduces the data in which the operator can understand and decide. For these kinds of applications there is a lot of processing and data gathering happens inside the embedded systems. The Manufacturer must invest a good amount to get these analytics data out for users from the embedded system. Few Customers buy the Application developed by the Manufactures to make use of the data to make business decisions.

However, few customers try other revere engineered applications developed by Hackers far less amount verses the Manufactures as manufacture do add the cost of the embedded system's data gathering work as well as the PC software. Because of this the manufacturer may lose on a lot of revenue as these features are software features and once done only require licensing to activate and have no additional tick cost which is revenue for them. To remove this risk from a business if such data is encrypted then it might be difficult for these hackers to easily decrypt the data and create the software

Following is a typical module diagram for an embedded system which is discussed in this paper. The proposed system shall be a drop-in module in an existing system and shall provide seem less API access to the application as provided by the existing CAN APIs also the configuration of the Encryption system shall be embedded in a system as other system configuration.

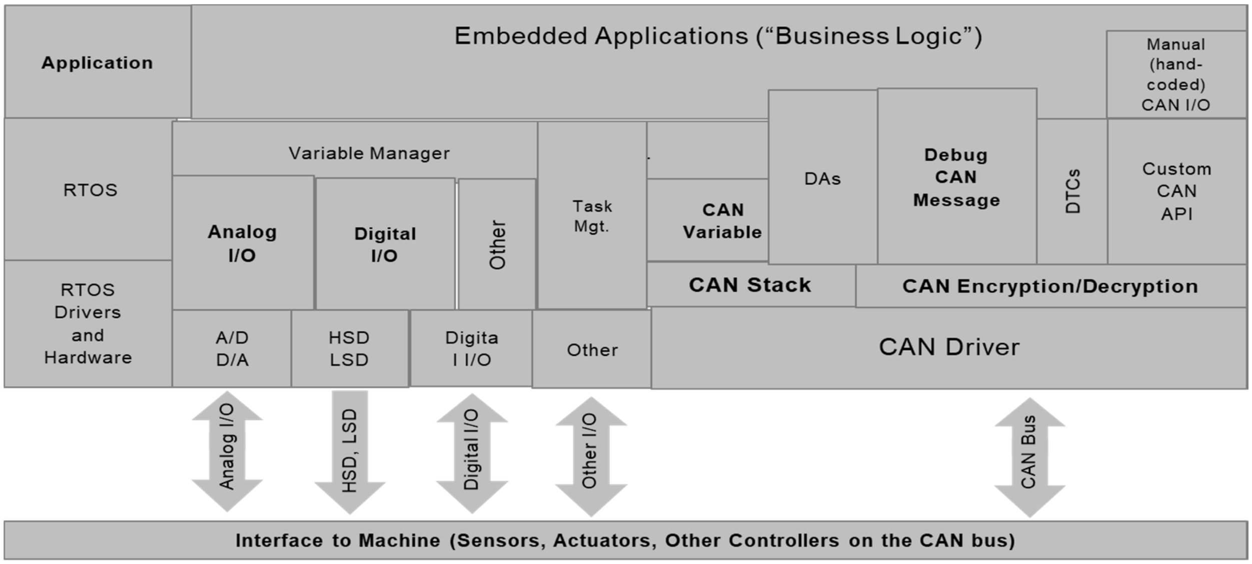


Figure 2 - Internal Embedded System with Modules

III. ENCRYPTION AND DECRYPTION

1) *Plain Text:* Plain Text is used to refer to a message before encryption or after decryption. It is always in a readable form. To provide security to it is converted into a non-readable form.
2) *Key:* A key is a string of bits. Those bits may be numeric or alphanumeric text or maybe a special symbol. In cryptography, Key plays an important role. At the time of encrypting the data key is used on plain text and at the time of decrypting the data it is used on the ciphertext. The same secret key for encryption and decryption of data. The security of data is fully dependent on the Key.
3) *Encryption:* It is a security purpose process. It is the process of converting the plain text into unreadable text using a secret key. For this it uses different encryption algorithms. It is the most efficient method to achieve data security by providing protection to the confidentiality of the message. For encrypting the data, a secret key is used. The data after encryption is called ciphertext and after decryption is called plain text.
4) *Cipher Data:* The data obtained after encryption is called as Cipher Data. It is unreadable. To convert it into a readable form is it given to the decryption process.

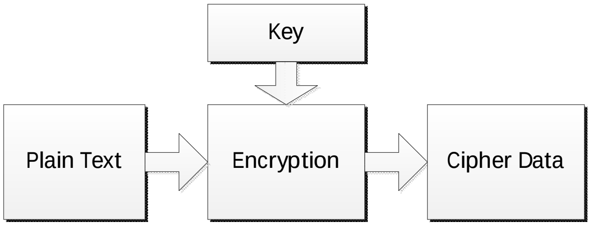


Figure 3 - Basic Encryption

5) *Decryption:* It is the process of taking encoded data and converting into original data. Decryption is used for un-encrypting the data with keys or algorithms. Cryptography uses the decryption technique at the receiver side to obtain the original message from a non-readable message (Cipher Data).

The decryption process can required Decryption algorithm and a key. A Decryption algorithm indicates the technique that has been used in Decryption. Usually, the encryption and decryption algorithm are the same only difference could the sequence of processing the data

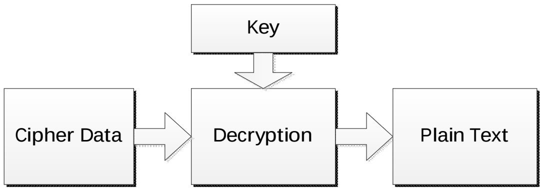


Figure 4 - Basic Decryption

IV. AES ALGORITHM

The AES stands for Advanced Encryption Standard (AES) algorithm. cryptography technique is used in the AES algorithm. Providing security to the data is important in data communication. So, to increase the security AES uses the Key Expansion algorithm in it. AES algorithm performs encryption of the data and decryption of the encrypted data. AES algorithm has a symmetric block cipher with a length of 128 bits. The encryption part of the algorithm gives the information to the mixed – up structure, after that information is called as a ciphertext. The decryption part gives the cipher content once again into its indigenous structure that is called as the original text. AES algorithms allows to use various length of key e.g 128 – bit, 192 – bit and 256 – bit. Here, a 128-bit key is used for encryption and decryption of the data. Each round will use the different 128 – bit key and that keys will be extracted from the original key using the AES algorithm. The 128-bit key is always arranged into 4*4 array matrix into 16 bytes that is a mirror quantity of 32-bit words in array. The number of words in the key is shown by Nk. This array is referred to as a State array in AES. The number of rounds performed during the execution of the algorithm is depending upon the key length and it is shown by Nr, where when Nk = 4 then Nr = 10, when Nk = 6 then Nr = 12 and when Nk = 8 then Nr = 14. As it is 128 – bit, the number of rounds will be 10. Each round of encryption and decryption consists of the 4 sub-process. - Byte substitution uses an S-box lookup table. - Row-wise stage of the State x - Column-wise blending inside every section of the State exhibit - Addition of round key to the State.

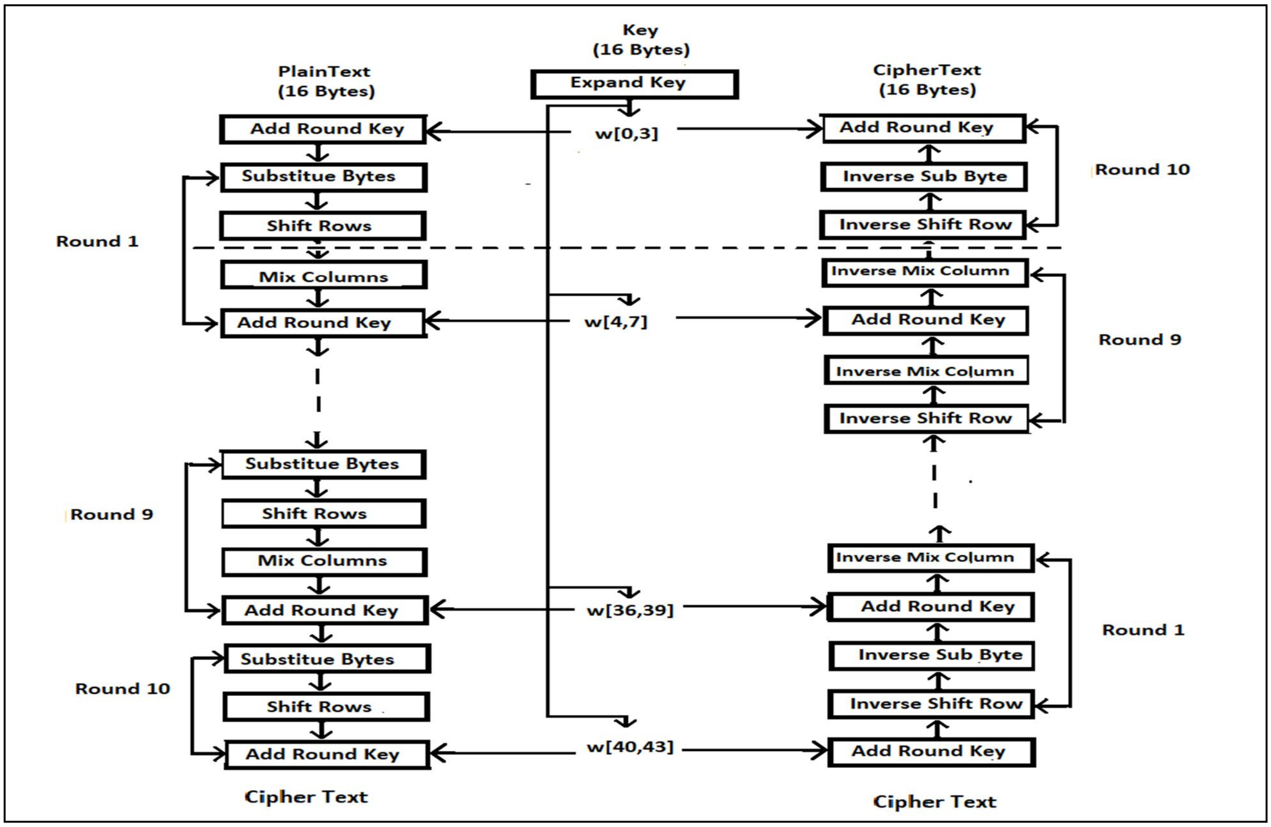


Figure 5 - Structure of AES

V. CAN ENCRYPTION AND DECRYPTION

A. CAN Encryption Intro

The configurable CAN Encryption system uses the following security features:

1) Configurable and customizable algorithms for
2) Generation and update of keys
3) AES-128/256 (Advanced Encryption Standard)
4) Checksum/hash calculation for authentication
5) Selective Encryption and decryption
6) All keys are synchronous and shared uniquely among two or multiple communication partners
7) The current key used is the dynamic key, which changes after a random time interval.
8) Permanent keys are used for initialization of the dynamic key which are stored in Non-Volatile memory
9) Support of a key hierarchy i.e. Serial number, License and Industry key

B. Encryption Group

The CAN Encryption grouping method connects a CAN configurator with a CAN device and provides a secure communication channel supporting both authentication and encryption. The pairing is essential in understanding in the system which all ECUs are ready to receive encrypted messages. The pairing can be divided into two parts one is static pairing which happens at the start of the system and the other is dynamic pairing. The static pairing can help us start the communication and the dynamic pairing helps the system make sure the external entity is not trying to sniff the data or request the key information under special circumstances camouflaging the system.

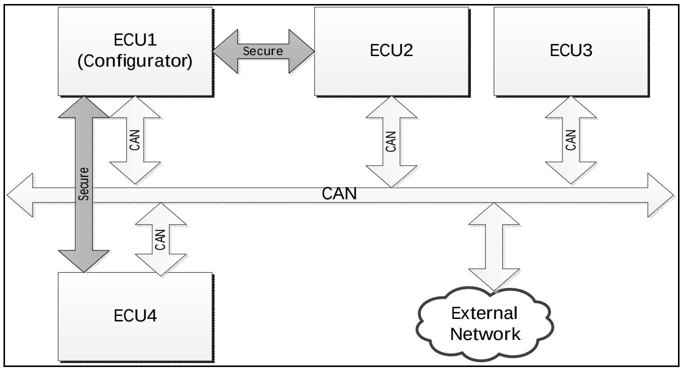


Figure 6 - Encryption Grouping

C. Key Generation

These Encryption Security systems require keys. Security keys require the Level of management so it is easy for a Manufacturer to work with it however shall make it difficult for an external entity to decipher this. This CAN Encryption System needs to have a key hierarchy. The Dynamic generation process includes multiple keys to help improve the uniqueness of the dynamic key generation. Multiple Keys helps generate higher the security level as the key Mix-up function have multiple combination to use.

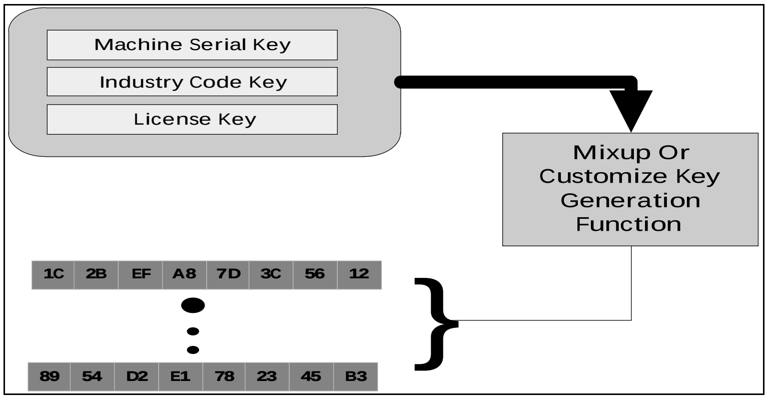


Figure 7 - Dynamic Key Generation

The Key generation as mentioned uses the hierarchal structure to Generate the key, the care shall be taken in how you Mix-up or Write this Customize function to generate these series of Keys and store them. This is a backbone of the system which makes it unique to add a further level of encryption. As this is a closely-knit security system, we can have these key generation at a central location and transfer only the keys to the respective devices or this code can run inside the logic to make on the go decision. In the current system we are proposing to this key generation at the central location and the keys can be pass as a configuration for the ECUs.

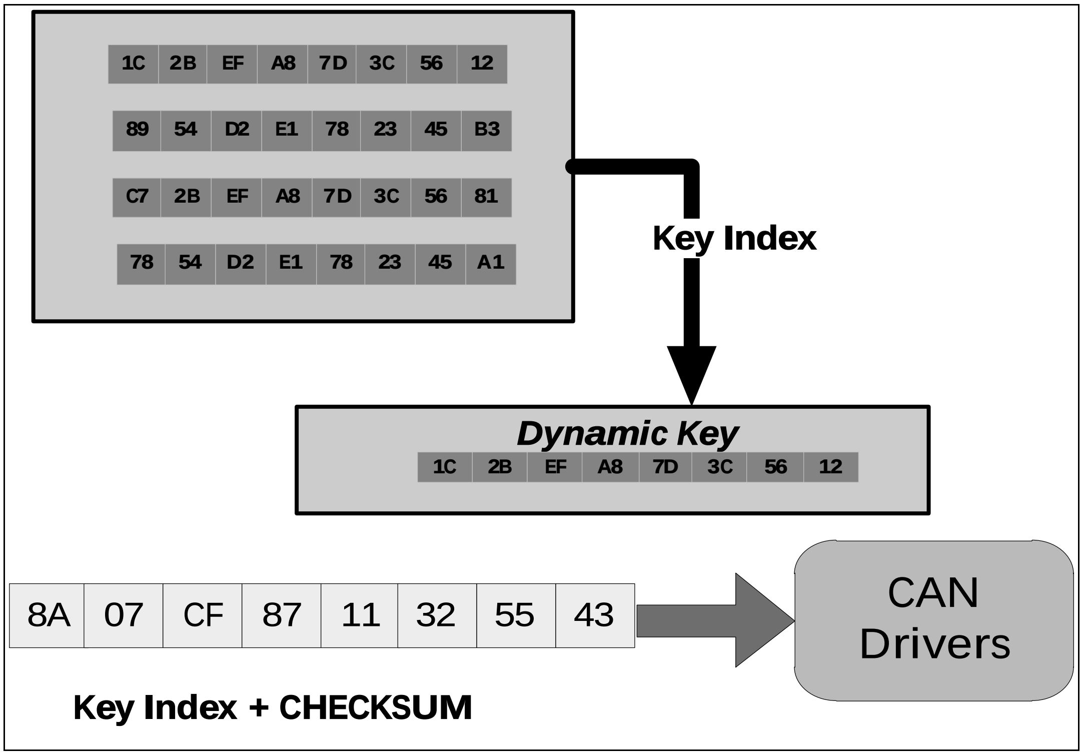
D. Index Key share

The Key generation process has static elements as mentioned earlier and stored in the system however at any point in time only one Key is used for encryption. As only one key is used by using reverse engineering it may be possible to decipher the key that's why the system proposed shall try to use multiple keys in random timeframes which makes this encryption robust. To mitigate the randomness the index key is shared as a reference.

The dynamic key gets continuously updated following a time scheme. As part of the secure synchronization, all grouped ECUs exchange random numbers. These shared random numbers are used to reference synchronized shared key.

E. Transmitting the Synchronization Message

The synchronization message has a combination of status byte and the encrypted message checksum. The synchronization status indicates the device is grouped actively and the communication is still authenticated. The following bytes in this message are a checksum of the data encrypted.



1C 2B EF A8 7D 3C 56 12

89 54 D2 E1 78 23 45 B3

C7 2B EF A8 7D 3C 56 81

78 54 D2 E1 78 23 45 A1

Dynamic Key

1C 2B EF A8 7D 3C 56 12

8A 07 CF 87 11 32 55 43

Figure 8 - Synchronization message

On receiving a secure synchronization message, a CAN Encryption ECU participates in the synchronization cycle by transmitting its own secure message and resetting its internal timer. Any device which receives these messages and detects an expiration of the receiving Sync event time starts the next secure synchronization cycle by transmitting its own secure synchronization message.

Any device active in encryption network can start the next synchronization cycle. For example, a device might want to have a received Synchronization message verified as fast as possible. However, the device must wait until the Set activation time has passed before starting a new cycle. This delay ensures that the CAN bus is not overloaded with synchronization messages.

F. Transmitting Encrypted message

Application for transmitting a secure message check the encryption message configuration. If the message to be transmitted is in the list of secure messages, a dynamic key is used, and a checksum is created. The encryption message contains control data in specific byte location including the CAN message ID as per secure list to follow and the current transmit message counter.

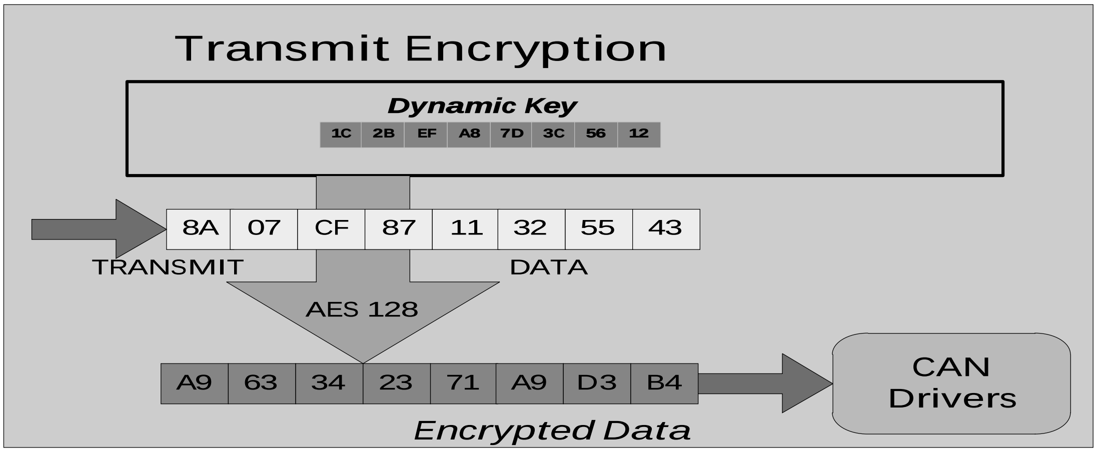


Figure 9 - Encrypted message transmission

The unique data is generated by calculating a checksum and encrypting it. Encryption happens based on the dynamic key generated and using AES.

G. CAN Decryption

1) *Receiving a CAN Encryption Synchronization:* On receiving a secure synchronization message, a device first decrypts the bytes based on the current shared key Index. Then the function calculates the checksum of the Synchronization message. If the checksum matches with the data received, the Message checksum is stored for further verification of encrypted data., once everything matches then the synchronization message is considered authenticated. On receiving a secure synchronization message, a function determines if the local Sync event activation time expired – if the time since the last transmission is greater than the Sync event activation time. If so, the device transmits its own next secure synchronization message.
2) *Receiving a Secured Message:* On the receiving side, if a message is received that is listed in the secure message list, the synchronization message is received first and stored in a buffer. The reception starts a timeout. A Synchronization that is received without a message following within a specified time of the random generator synchronizes at the transmission and reception end then is considered an error, and a security error counter gets incremented. The included message sequence counter is checked. The sequence counter contains information about the dynamic Sync cycle and used to determine the current dynamic key or the previously received one gets used.

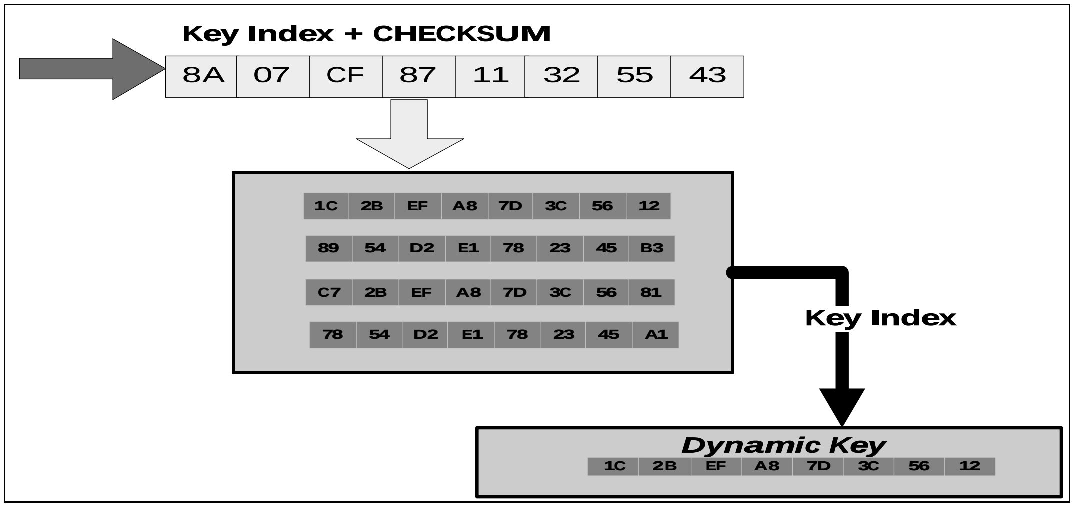


Figure 10- Encryption Synchronization received

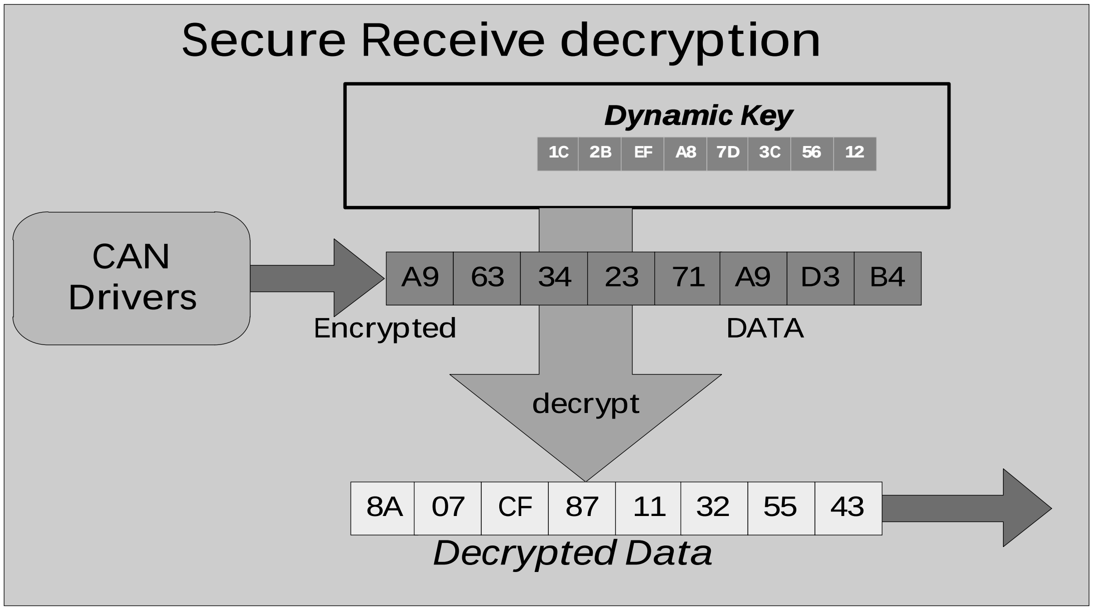


Figure 11 - Decrypt message received

Once the dynamic key determined based on an earlier synchronization message key index then the message is decrypted using the dynamic key and AES decryption. Later the data check decrypted is verified with the checksum send in the synchronization message to verify data integrity.

A secure message received is passed on to the application or protocols above the CAN Encryption handler only if authentication was successful. Otherwise the received message does not get passed on to the application and discarded. The Discard message counter has been deployed to make sure possible intrusion.

As the current system does have a possible intrusion detection based on Synchronization message and checksum verification however do not employ any mechanism to shut the encryption process or adapt to new keys. In the future this is one more possibility to have adoptive encryption. The current system needed some definite time reliability, so these new measures are not employed**.**

VI. CONCLUSION

The proposed secure communication system for Embedded Application uses the Standard AES128/256 encryption or Any other standard Encryption based on the amount of data to be encrypted and the Performance of the overall system. This system is unique in a way of using multiple dynamic keys and not sharing the dynamic keys over CAN communication instead share a pointer to get to this dynamic key. This makes the proposed system robust. In addition to the dynamic keys the randomness is added when the dynamic keys are used to make this system even more secure. Also, the first-time authentication of grouping, as well as the dynamic authentication, make sure no other system is mimicking the behavior of the secure ECUs to decipher keys.

The Integration of the Secure communication system in existing embedded application modules is easy and configurability makes this process seem-less for users to adopt. There are further security possibilities however this system might deter hackers to replicate the application as it may need a lot more engineering efforts to extract data out of the encrypted messages.

REFERENCES

[1] “Implementing scalable CAN security with CANcrypt, Authentication and encryption for CANopen, J1939 and other Controller or CAN FD protocols”

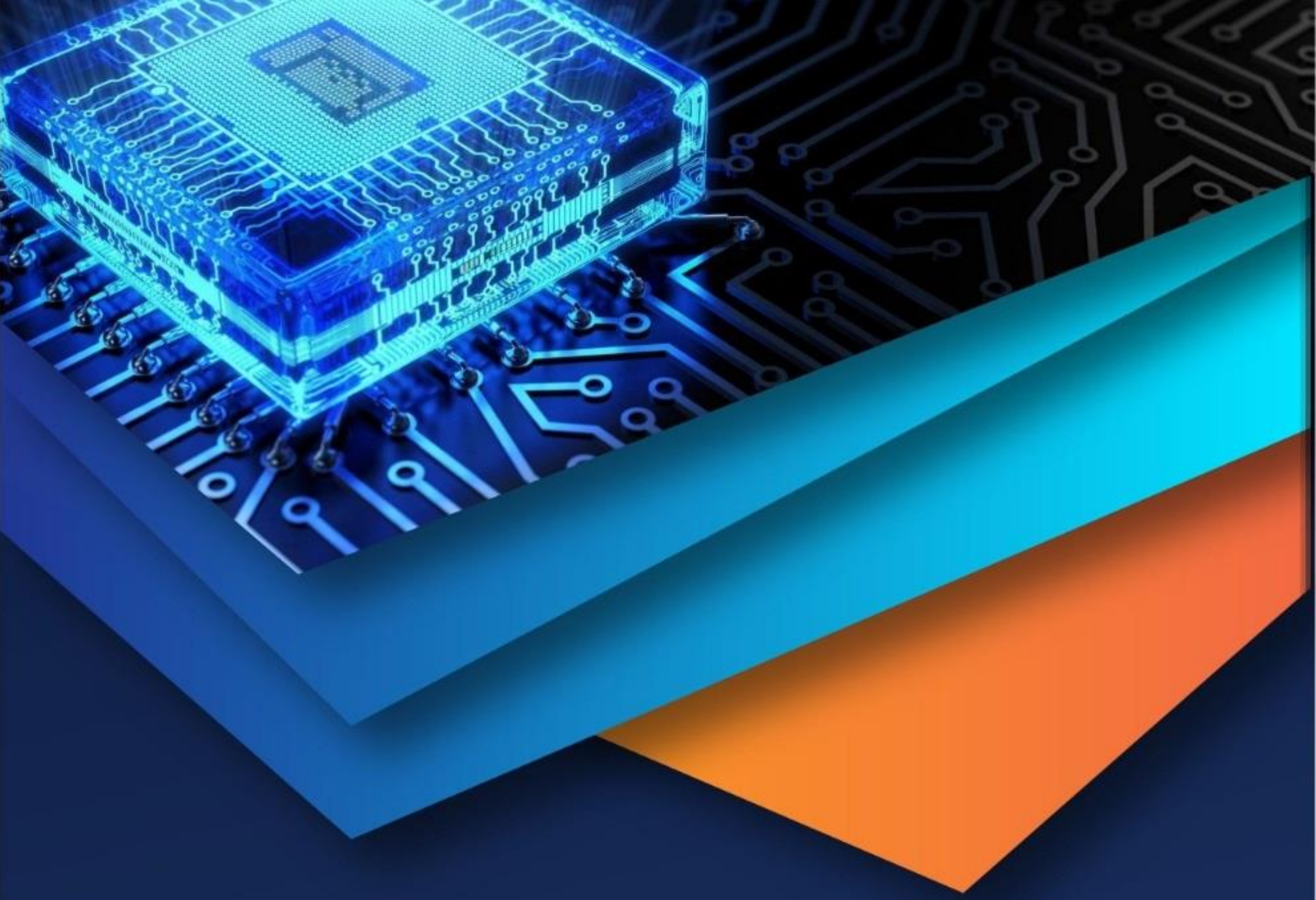
[2] International Journal of Advanced Research in Computer Science “Data Encryption Using Different Techniques: A Review”

[3] Ankid Fadia, Jaya Bhattacharjee, “Encryption, Protecting Your Data”, Vikash Publishing House Pvt Ltd,2007, ISBN: 812592251-2

[4] H. Krawczyk, “The Order of Encryption and Authentication for Protecting Communications”, http://eprint.iacr.org/2001. 2014 international Conference on Devices, Circuits and Communications (ICDCCom)

[5] “Controller Area Network Primer” documents like www.computer-solutions.co.uk/download/Peak/CAN-Tutorial.pdf

[6] R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. IEEE Trans. Plasma Sci. [Online]. 21(3). pp. 876–880. Available: http://www.halcyon.com/pub/journals/21ps03-vidmar



10.22214/IJRASET

45.98

IMPACT FACTOR:
7.129

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY