논문 18-43-09-16 The Journal of Korean Institute of Communications and Information Sciences '18-09 Vol.43 No.09
https://doi.org/10.7840/kics.2018.43.9.1525

보안 알고리즘을 반영한 확장된 CAN Bus 통신에 관한 실증적 연구

홍 봉 조◆, 한 인 철*, 장 동 원*, 이 남 용**

Empirical Study on the Extended CAN Bus Communication with Security Algorithms

Bong-Jo Hong◆, In-Chul Han*, Dong-Won Jang*, Nam-Yong Lee**

요 약

자동차 컨트롤유닛 프로세서 간 통신과 엘리베이터와 같은 산업기기에 많이 사용하는 CAN Bus 통신은 BOSCH(사)가 제안한 통신방식으로 암호화 없이 대부분 사용하고 있다. 4차 산업 혁명에서의 자율주행 자동차 프로세서 간 통신인 CAN Bus 통신이 암호화되지 않았다면 외부로부터 무단으로 접속되는 경우 해킹의 피해로 이어질 수 있다. 따라서 CAN Bus 통신의 보안성을 높이기 위해 통신데이터를 암호화하여야 한다.

본 논문에서는 Embedded RTOS 환경하에서 CAN Bus 통신 System을 구현하여 AES, ARIA, HIGHT 등과 같은 표준화된 암호화 방식별로 암호화 시간과 복호화 시간, Bus Load, 통신 성능 등을 비교 측정 분석하고 통신 성능을 높일 방법을 제시한다.

Key Words : can Bus, AES, ARIA, HIGHT, FreeRTOS

ABSTRACT

Automobile control unit CAN bus communication, which is widely used in industrial devices such as interprocessor communication and elevator, is a communication method proposed by BOSCH, and is mostly used without encryption. Autonomous driving in the fourth Industrial Revolution Unless the CAN bus communication between automobile processors is encrypted, unauthorized access from the outside can lead to hacking damage. Therefore, communication data must be encrypted to increase the security of CAN bus communication.

In this paper, we implement a CAN Bus communication system under embedded RTOS environment, and propose a method to compare and measure the encryption time, decoding time, bus load, communication performance, and communication performance according to standardized encryption methods such as AES, ARIA and HIGHT.

Ⅰ. 서 론

CAN(Controller Area Network) Bus 통신 프로토콜은 BOSCH(사)가 제안한 통신 방식으로 차량을 비롯하여 많은 산업기기에 사용되고 있으며, 보안이 없이 대부분 사용하고 있다[1]. 무인 자율 주행 자동차 혹

◆ First Author : (ORCID:0000-0001-6566-1940)Soongsil University Graduate of IT Policy and Management, hongbongjo@gmail.com, 정회원
* (ORCID:0000-0001-9756-1397, 0000-0002-9639-4619)숭실대학원 IT정책경영학과, chulinh@hanmail.com, jangdongwon@hanmail.net
** 숭실대학교 소프트웨어학부 교수, nylee@ssu.ac.kr, 정회원
논문번호 : 201807-0-199-SE, Received July 2, 2018; Revised July 30, 2018; Accepted July 30, 2018

은 엘리베이터 등의 기기가 보안 없이 통신할 경우 해킹에 노출되어 있기 때문에 암호화를 하면 해킹위험을 줄일 수 있다[8].

CAN Bus 통신 표준규격은 ISO11898이며, 한 번에 보낼 수 있는 Data는 8 Byte로 미국표준인 AES 암호화 방식과 한국표준인 ARIA 방식으로 암호 통신을 할 경우 최소 Data가 16 Byte가 되기 때문에 한 번에 전송할 수 없다. 이로 인하여 암호화 통신할 경우 성능이 감소하게 되는데, 암호화 방식별로 통신성능을 비교 분석한다.

32 Bit CPU와 CAN Controller Device로 검증용 Hardware를 구현하고, FreeRTOS 환경하에서 ISO11898 Protocol을 구현하여 성능시험용 알고리즘을 제작하고, AES 암호방식과 ARIA 암호방식, 64bit HIGHT 등 3가지의 암호화에 대한 통신 성능은 평문일 때, AES 128비트, 192비트, 256비트, ARIA 128비트, 192비트, 256비트 방식별로 각각 같은 길이의 암호화 패킷을 보내 수신 측에서 Bitrate 시간으로 비교 분석하며, 암호화와 복호화에 처리 시간은 암호화 방식별로 암호화 시작 지점과 완료 지점에 Hardware Flag를 두어 Oscilloscope 계측 장비로 시간을 측정한 후 어떤 방식이 가장 효율이 높은지 비교 분석하여 암호화에 적합한 CAN 통신 알고리즘을 제안한다.

Ⅱ. 관련연구

2.1 CAN Bus 통신

CAN Bus 통신은 저비용으로 최대 1MBit/s까지 사용할 수 있으며 Multi Node 구성된 자동차 엘리베이터 등에 많이 쓰인다. Address Bit 수에 따라 11 Bit 식별자를 가진 CAN 2.0A Protocol과 그림 1.과 같이 29 Bit 식별자를 가진 CAN 2.0B 방식이 있다[1,2,6]. 본 연구에서는 이를 반영하여 화장된 CAN Bus 통신 Protocol을 제안하였다.

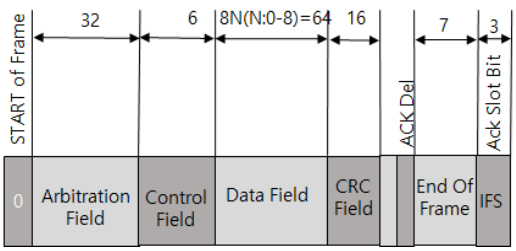

Fig. 1. CAN 2.0B Packet Structure

2.2 AES 암호

AES(Advanced Encryption Standard)는 미국표준연구소(NIST)에서 2001년도에 제정된 암호방식으로 1977년 공표된 DES를 대체한 AES는 암호화 복호화 과정에서 같은 KEY를 사용하는 대칭키 알고리즘이다. AES의 특징은 안전성(Security), 저비용(Cost), 알고리즘 및 구현 특성이며, KEY는 128 비트, 192 비트, 256 비트로 확장 가능하며, 가장 널리 사용하는 미국 정보표준 암호화 알고리즘이다[3].

2.3 ARIA 암호

ARIA(Academy Research Institute Agency)는 2004년 12월에 한국산업규격 KS표준으로 제정된 것으로 대칭키 알고리즘이며 순수 국내 기술로 개발된 알고리즘이다. 블록 크기는 128 비트이며 128 비트, 192 비트, 256 비트의 확장키를 사용할 수 있다. ARIA의 특징은 경량 환경의 하드웨어에서 높은 효율성에 있으며, SEED 알고리즘보다 빠른 성능을 제공하고 있다. 객관적인 안정성 및 효율성 평가를 위하여 NESSIE(New Europen Schemes for Signature Integrity, and Encryption) 의 주관 기관인 벨기에 루벤대학에 의한 평가를 받았다[4].

2.4 HIGHT 암호

HIGHT(HIGH security ans light weigHT)는 RFID, USN 등과 같이 저전력, 경량화를 요구하는 컴퓨팅 환경에서 기밀성을 제공하기 위해 2005년 KISA와 고려대학교가 공동으로 개발한 64 비트형 블록 암호 알고리즘이다. 알고리즘의 전체 구조는 일반화된 Feistel 변형 구조로 이루어져 있으며, 64 비트의 평문과 128 비트 카로부터 생성된 8개의 8비트 화이닝 키와 128개의 8비트 서브키를 입력으로 사용하며 총 32라운드를 거쳐 64 비트 암호문을 만든다[5].

Ⅲ. CAN Bus 통신보안시스템 구성

3.1 CAN Bus System

그림 2.와같이 96MHz 32 Bit ARM CPU와 Microchip사의 CAN Bus Controller MCP2515를 이용하여 암호통신 성능분석용 System Module 2 Set를 개발한다. MCP2551을 이용하며, CPU와 MCP2515는 SPI 통신으로 연결한다. CAN Module 1과 CAN Module 2는 1쌍의 CAN Bus로 서로 연결하고, PEAK system(사)의 PCAN-Explorer6와 PCAN-View를 연결하여 CAN Bus의 Load와 Bitrate 을 검증한다.

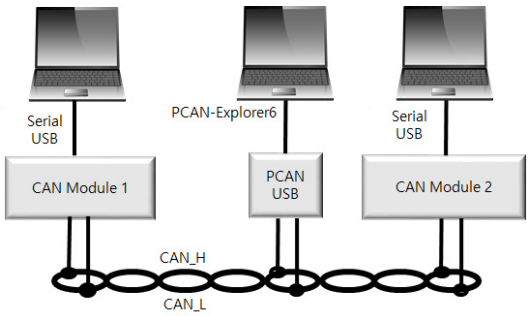

Fig. 2. Configuration of CAN Bus communication system

3.2 CAN Bus 통신 Hardware & Software

CAN Bus 통신의 Hardware Block은 그림 3.과 같이 CPU와 CAN Controller, Line 보호용 Surge Protector, USB Serial Interface를 하기위한 Block, 전원을 공급하기 위한 DC to DC Switching Regulator Block으로 구성된다.

CAN Bus 암호통신을 위한 Software Block은 그림 4.와같이 FreeRTOS 환경하에 Packet Generation Task, Packet Receive Task, Bitrate Analysis Task, USB Serial Task, AES/ARIA/HIGHT Module, MCP2515 CAN Driver Module로 구성한다. CAN Driver Module과 CAN Packet 송신을 하기위한 CAN Packet TX Queue와 수신된 Packet을 전달하기 위한 CAN Rx Queue를 둔다.

그림 5.와 표 1.에서처럼 16 Byte 전송의 Sequence chart Diagram처럼 Packet 전송 명령이 오면

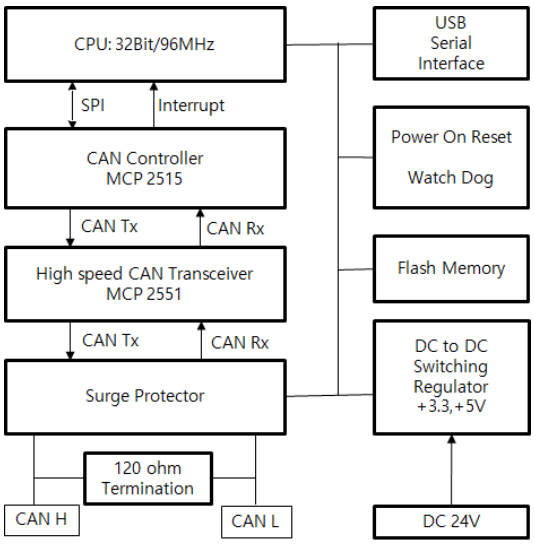

Fig. 3. CAN Bus Communication Hardware Block

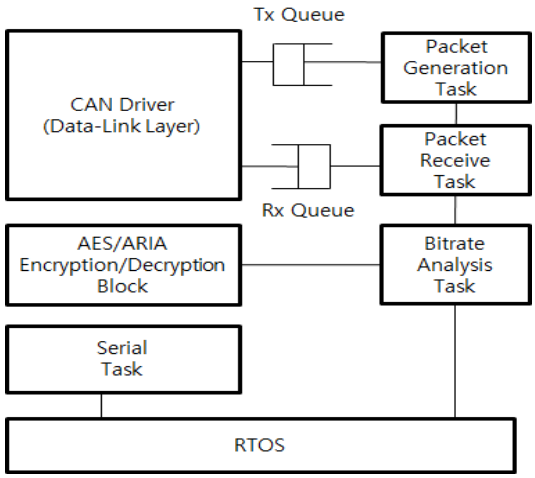

Fig. 4. CAN Bus Communication Software Block

AES128, AEA192, AES256, ARIA128, ARIA192, ARIA256, HIGHT등의 암호 방식으로 Sequence No와 Checksum으로 이루어진 Packet을 암호화한 후 8Byte의 Packet을 CAN Driver Module로 송신을 한다.

수신 측에서는 CAN Driver에서 Interrupt가 발생하면 SPI 통신으로 8byte의 CAN Packet Data를 가져와서 저장한 후 두 번째 Packet을 기다린다. 송신 측에서 나머지 8byte를 보내게 되면 수신 측에서 추가로 8byte를 받아서 복호화하게 된다. 복호화 할 때 어떤 암호화 방식을 사용하였는지 알고 있어야 복호화가 가능하다. AES 방식과 ARIA 방식은 2개의 Packet을 받아 각각 평문으로 만든 다음 Sequence 번호와 Checksum data를 보고 정상적인 Packet이 수신된 경우 Bitrate 을 계산하여 Serial Monitor에 출력한다.

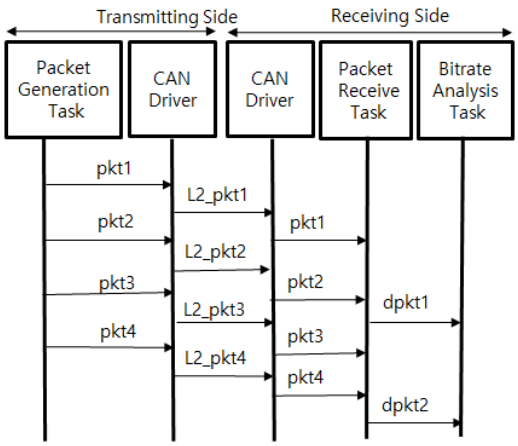

Fig. 5. CAN Communication Sequence Chart Diagram

Table 1. Sequence chart Diagram Message

No.	Message	Message 설명
1	pkt1 pkt3	The first encrypted Sequence No and Checksum
2	pkt2 pkt4	Second Encrypted Sequence No and Checksum
3	L2_pkt1 L2_pkt3	First Packet in CAN Layer2 format
4	L2_pkt2 L2_pkt4	Second Packet in CAN Layer2 format
5	dpkt1 dpkt2	Plain Packet

CAN Bus의 표준은 그림 6.처럼 ISO 11890에 정의되어 있다. ISO 11890은 Physical Layer, Data-Link Layer, Application Layer로 구성되어 있다. Application Layer에서 암호화와 복호화 Block, Bitrate 분석 등의 Software Module이 있으며, Data Link Layer에는 CAN Driver Software Module이 있다.

송신 측 Packet 전송의 과정은 그림 7.에서 설명한다. 매 Packet은 1ms 주기로 평문에 Sequence 번호와 Checksum Data를 포함하여 암호화한다. AES 방식과 ARIA 방식은 암호화 Packet이 최소 16 Byte가 되므로 두 번에 나누어 보낼 수 있다. 첫 번째와 두 번째 보낼 Packet을 송신 Queue에 넣고, CAN Controller에서 송신 Reg.를 보아 송신 가능 상태가 되면 8 Byte

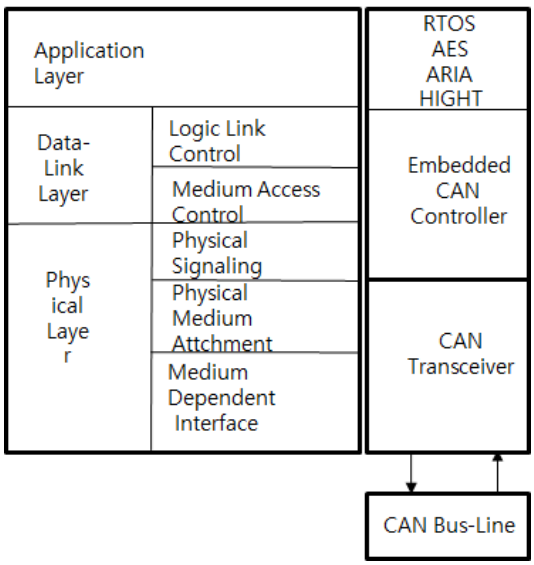

Fig. 6. CAN Bus Communication Protocol Stack

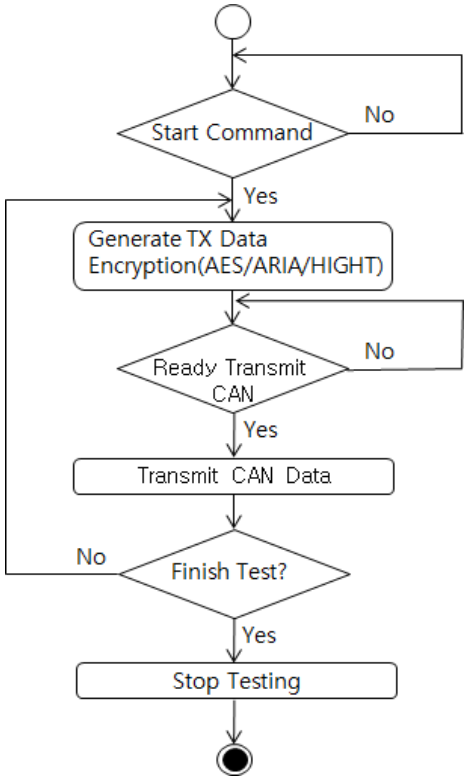

Fig. 7. Sender Activity Diagram

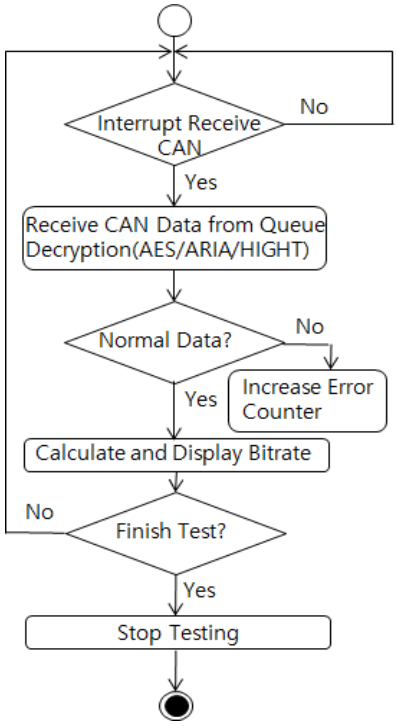

Fig. 8. Receiver Activity Diagram

단위로 Packet을 보낸다.

Packet을 수신하는 과정은 그림 8.의 Activity Diagram에서 보여준다. 송신 측에서 Packet을 전송하면 CAN Controller에서 CAN Packet을 받아 Packet이 정상이면 CPU에 Interrupt를 발생하게 된다. Interrupt가 발생하면 SPI 통신을 이용하여 CAN Controller에 저장된 수신 Packet을 수신하여 Buffer에 저장하고 Data의 Checksum을 계산하여 정상적인 Data면 두 번째 Packet을 기다린다. 만약 두 번째 Packet도 정상적이면 두 개의 Packet으로 복호화 과정을 거친다. 복호화한 후에 매초 동안의 받은 Packets와 CAN Bit 수로 Bitrate를 계산하여 출력한다.

Ⅳ. 비교분석

4.1 암호화 처리시간

암호화 처리시간을 측정해본 결과 그림 9.에서 보는 바와 같이 ARIA 방식이 AES 방식보다 8배 이상 빠른 성능을 보였으며, HIGHT ECB 방식은 1ms 정도로 ARIA보다 2.5배 정도 더 소요되었다.

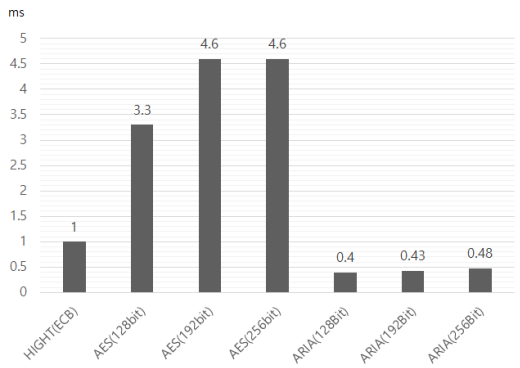

Fig. 9.Encryption processing time

4.2 복호화 처리시간

복호화 처리시간은 그림10.에서 보여주며, 암호화 시간보다는 작게 걸린다. 복호화 시간 또한 AES 방식이 ARIA 방식보다 10배 이상 걸린다.

4.3 CAN Bus Load Mean Value

그림 11.에서처럼 CAN Bus Load는 CAN Clock 속도가 125K일 때는 평문 전송 시 18.5%이며, 50Kbps 일 때는 70.6%이다.

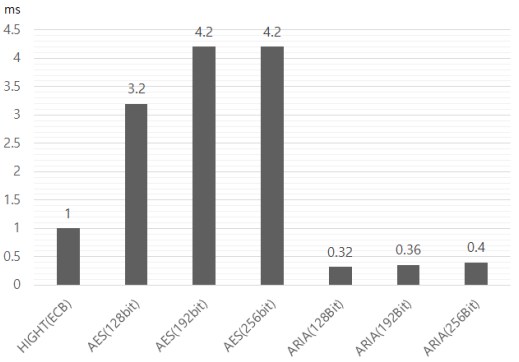

Fig. 10. Decryption processing time

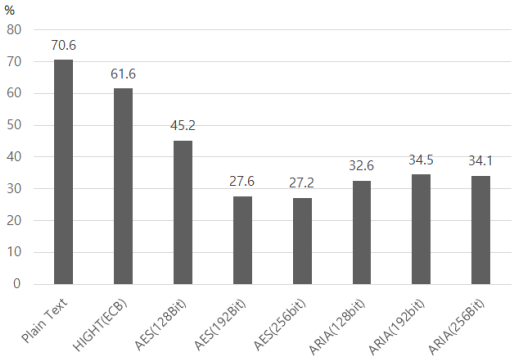

Fig. 11. CAN Bus Load Mean Value

4.4 암호화 방식별 통신속도 비교

그림 12.에서처럼 평문일 때는 32,046 Bitrate, HIGHT 방식은 29,539 Bitrate, AES128 Bit 10,791 Bitrate, ARIA128 Bit는 14,497 Bitrate로 AES 방식보다 ARIA 방식이 134% 더 효율이 높았고, ARIA128 bit 방식이 HIGHT 방식보다 49% 정도 효율이 낮았다. 128 Bit 방식이 64 Bit 방식보다 낮은 이

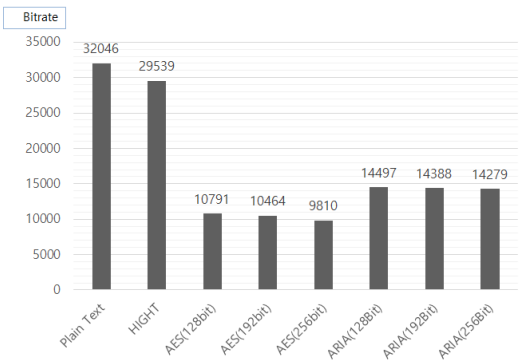

Fig. 12. Comparison of communication speed by encryption method

유는 암호화된 최소 Packet 길이가 128 Bit이기 때문에 한 번에 보낼 수 없어 두 번에 걸쳐 보내는데 많은 Overhead가 들어가기 때문이다. 따라서 통신 효율을 높이려면 한 번에 최소 16 Byte 길이의 Packet을 보낼 수 있는 구조이어야 한다. CAN 2.0B Frame에서 Data Length Code가 현재는 8Byte까지 표시되는데, 16Byte 까지 사용할 수 있도록 하고, Data Field는 16N(N:0-16)=128bit로 하여 길이를 확장하여야 암호화된 Packet인 경우 통신 성능을 높일 수 있다.

$$\text{Bitrate}=(\text{수신Packet} * 10) * 8 + (\text{수신Packet} * 29)(\text{단위:bps}) \quad (1)$$

4.5 암호화 방식별 비교

CAN Bus 통신에 암호화를 하는 경우 암호화 방식에 따라 성능에 차이를 보였다. 이러한 차이를 CAN 통신 System Module을 통하여 측정하였으며, PCAN-Explorer6 계측 Tool로서 상호 검증한 결과 CAN Bus 통신 암호화 처리에 걸리는 시간은 HEIGHT 방식 1ms, AES 128비트3.3ms, ARIA 128 비트 0.4ms 가 측정되었다. CAN Bus 통신 복호화 처리에 걸리는 시간은 HEIGHT 방식 1ms, AES 128비트 3.2ms, ARIA 128비트 0.32ms 가 측정되었다. 통신 속도 분석결과 평문일 때 32,046 bps HEIGHT 29,539 bps, AES 128비트 10,791 bps, ARIA 128비트 14,497 bps로 측정되었다.

ARIA 128 비트는 14,497 Bitrate로 AES 방식보다 ARIA 방식이 134% 더 전송효율이 높았고, ARIA 128 비트 방식이 HIGHT 방식보다 49% 전송효율이 낮았다. AES 방식과 ARIA 방식은 최소 암호화 패킷 길이가 16 Byte가 되어 현재의 CAN Bus 표준 방식으로 데이터를 보내려면 2번에 나누어 보내야 한다. 통신 효율은 2번을 보낸 후 다시 모아서 복호화하는 과정에서 통신성능의 저하를 가져 왔다. HEIGHT 방식은 평문을 암호화했을 경우 8byte이기 때문에 AES 암호방식이나 ARIA 암호방식보다 효율이 높았다.

V. 결 론

본 논문은 Embedded RTOS 환경에서의 CAN Bus 암호화 통신 성능을 측정하기 위하여 96MHz 32 Bit ARM CPU와 Microchip 사의 CAN Bus Controller MCP2515를 이용하여 CAN Bus 시험용 시료를 제작한 후 암호화 방식별로 암호화시 처리시간, 복호와 처리시간, AES 128비트, 192비트, 256비트, ARIA 128 비트, 192비트, 256비트, EIGHT 64비트에 대한 통신시험을 실시 해본 결과, 전반적으로 ARIA 방식이 AES 방식보다 통신 및 암호화, 복호화 처리 효율이 높았다. 가장 전송효율이 높게 나타난 것은 HIGHT 64비트 방식이었고, 64비트의 HIGHT 암호방식이 효율이 높은 것은 최소 암호화된 Packet이 8 Byte이기 때문이다. 하지만 암호 안정성이 최소 128비트 이상의 KEY 비트를 필요로 함으로 사용 시 주의가 필요하다. AES, ARIA 암호화로 CAN 통신에 사용하려면 한 번에 8 Byte인 현재 국제표준인 CAN 2.0B 규격에서 Data Field 부분을 16byte 이상으로 사용하는 새로운 규격을 제안한다. CAN 2.0B 규격에서 한 번에 16 Byte 이상 보낼 수 있는 CAN Controller로 AES 및 ARIA 암호방식의 통신효율에 관한 추가적인 연구가 필요하다.

References

[1] O. Avatefipour and H. Malik, "State-of-the-art survey on in-vehicle network communication "CAN-Bus" security and vulnerabilities," *Int. J. Comput. Sci. and Netw.*, 2017.

[2] R. Došek, et al., "Secure high level communication protocol for CAN BUS," *Annals of DAAAM & Proc. 2015*, vol. 26 no. 1, pp. 1009-1015, 2015.

[3] D. Selent, "Advanced encryption standard," *Rivier Academic J.*, vol. 6, no. 2, pp. 1-14, 2010.

[4] KISA, "Combined public-private-use block cipher algorithm ARIA algorithm specification," pp. 4, 2004

[5] KISA, "HIGHT block encryption algorithm specification and detailed specification," pp. 2, 2009.

[6] K. W. Kang, "Real time framework and platform design based on CAN communication," Ph.D. dissertation, Dept. of Compt., Chonnam Univ., 2015.

[7] S.-R. Yeom, "A study of the security enhancements of smart car through the access restrictions of the CAN bus," M.S. Thesis, Dept. of College of Inf. Technol., Soongsil Univ., 2016.

홍 봉 조 (Bong-Jo Hong)

1986년 2월 : 숭실대학교 전자공학과 졸업(학사)
2017년 6월 : 숭실대학교 정보과학대학원 공학석사
1917년~현재 : 숭실대학교 IT정책경영학과 박사과정
<관심분야> 정보통신, IOT, 정보보호, Embedded System

장 동 원 (Dong-Won Jang)

2002년2월 : 남서울대학교 정보통신공학 졸업
2006년 2월 : 숭실대학교 정보과학대학원(소프트웨어전공) 석사
<관심분야> 스마트시티, 영상보안

한 인 철 (In-Chul Han)

1984년 2월 : 고려대학교 건축학과 졸업
2003년 8월 : 헬싱키 경영대 학원(Aalto) 석사
<관심분야> 통신보안, 블록체인

이 남 용 (Nam-Yong Lee)

1983년 : 고려대학원 경영대학원석사 졸업
1993년 : 미국미시시피주립대 박사 경영학박사
<관심분야> SW테스트, 품질보증, MIS, 정보보호 등이다.