https://doi.org/10.21289/KSIC.2017.20.2.177 KSIC

외부 해킹 방지를 위한 CAN 네트워크 침입 검출 알고리즘 개발

Development of CAN network intrusion detection algorithm to prevent external hacking

김현희[1], 신은혜[2], 이경창[1], 황용연[1*]

Hyun-Hee Kim[1], Eun Hye Shin[2], Kyung-Chang Lee[1], Yeong-Yeun Hwang[1*]

〈Abstract〉

With the latest developments in ICT(Information Communication Technology) technology, research on Intelligent Car, Connected Car that support autonomous driving or services is actively underway. It is true that the number of inputs linked to external connections is likely to be exposed to a malicious intrusion. I studied possible security issues that may occur within the Connected Car. A variety of security issues may arise in the use of CAN, the most typical internal network of vehicles.

The data can be encrypted by encrypting the entire data within the CAN network system to resolve the security issues, but can be time-consuming and time-consuming, and can cause the authentication process to be carried out in the event of a certification procedure.

To resolve this problem, CAN network system can be used to authenticate nodes in the network to perform a unique authentication of nodes using nodes in the network to authenticate nodes in the nodes and By encoding the ID, identifying the identity of the data, changing the identity of the ID and decryption algorithm, and identifying the cipher and certification techniques of the external invader, the encryption and authentication techniques could be detected by detecting and verifying the external intruder.

Add a monitoring node to the CAN network to resolve this. Share a unique ID that can be authenticated using the server that performs the initial certification of nodes

1* 정회원, 교신저자, 부경대학교 제어계측공학과 (yyh@pknu.ac.kr)
2. ㈜퓨트로닉

1* Corresponding Author, Department of Control and Instrumentation Engineering, Pukyong National University
2. pyuteuronik CO., LTD

within the network and encrypt IDs to secure data. By detecting external invaders, designing encryption and authentication techniques was designed to detect external intrusion and certification techniques, enabling them to detect external intrusions.

Keywords : Connected-car, CAN, intrusion detection, vehicle network, ECU

1. 서론

전기전자 기술이 발전하면서 차량에도 전자식 연료분사장치, ABS(Anti-Lock Braking system), ACC(Adaptive Cruise Control), 차량 자세 제어장치와 같은 전자 제어장치(ECU, Electronic Control Unit)가 다양하게 탑재되고 있다. 이러한 차량용 전자제어 장치는 CAN (Controller Area Network), LIN(Local Interconnect Network), FlexRay, MOST(Media Oriented System Transport) 등의 프로토콜을 이용하여 상호 통신한다[1].

최근까지 차량 통신은 차량 내부용 통신으로 주로 사용되었지만, 도로 인프라와 융합된 지능형 교통 시스템(C-ITS, Cooper ative Intelligent Transport System), 자율주행 차량, 텔레매틱스(Telemetics) 서비스를 제공하는 지능형 자동차에 대한 연구가 활발히 진행되면서 차량 내외부 통신에 대한 연구도 활발히 진행되고 있다[2].

즉, 차량의 지능화, 정보화가 빠르게 진행되면서 악성 컴퓨터 바이러스와 같이 외부 해커 침입 등에 대한 문제가 발생될 수 있다. 이는 차량 ECU에 인위적으로 접근하여 차량의 정상적 주행을 방해하거나 오작동을 유발시켜 심각한 차량 사고로 이어질 수 있다[3].

차량의 사고율을 낮추고 운전자에게 다양한 편의를 제공하기 위해 학계와 자동차업체 등을 중심으로 자동차 IT 융합기술에 대한 연구가 활발히 진행되고 있다. 그러나 차량 네트워크 보안기술이 뒷받침되지 않는다면 차량 피해 뿐 아니라 운전자의 생명까지도 위협할 수 있다. 따라서, 커넥티드 카(connected-car) 시대를 대비하기 위해서는 차량을 외부해킹으로부터 보호할 수 있는 차량용 네트워크 보안 시스템에 대한 연구가 필요하다[4]-[6].

이를 위해, 본 논문에서는 커넥티드 카를 위한 차량용 네트워크 침입 탐지 검출 알고리즘에 대해 제안하고자 한다. 또한, 차량 ECU의 부하를 최소화하면서 침입 탐지 알고리즘을 개발하기 위하여 최소한의 하드웨어 구성과 알고리즘을 이용한 시스템을 제안하고자 한다.

이를 위해 본 논문은 서론을 포함하여 총 4장으로 구성하였다. 2장에서는 기존 연구와 차별화된 본 논문의 시스템 설계 및 구현에 대해서 기술하고, 3장에서는 이에 대한 구현과 성능 평가 결과를 기술하고 4장에서 결론을 맺도록 한다.

2. CAN 네트워크 침입 탐지 검출 시스템의 구조

본 논문에서 제안하는 차량 침입 탐지 시스템의 개념도는 그림 1과 같다. 제안 시스템은 동일 네트워크로 구성된 CAN 기반 ECU들의 인증과 모니터링을 담당하는 서버 노드인 MSU(Monitoring Server Unit)의 추가가 가장 큰 특징이다. 이 모니터링 서버 노드를 통해 접속하는 노드들의 접속여부를 확인하고 전체 노드가 모두 접속한

경우 CAN 통신을 시작하도록 구성한다. 접속하는 ECU는 BMS(Batter Management System), MCU(Motor Control Unit), VCU(Vehicle Control Unit), OBC(On Board Charger) 등의 노드로 구성 가능하다.

MSU의 경우 동일 네트워크상에 접속해야하는 모든 노드를 알고 있고, 각 노드들은 각 수신하는 CAN ID들의 송신 노드 정보를 알고 있어야 한다. 외부 침입이 탐지된 경우, 각 노드는 MSU로 알림을 보내 차량 동작에 관여하도록 한다. 이러한 방식은 모든 CAN ID를 MSU에서 확인했을 때 발생할 수 있는 노드 통신 부하 및 연산 부하를 줄이고, 각 기능을 분산할 수 있도록 하기 위함이다.

2.1 접속노드 인증 방법

본 논문에서 외부 침입을 탐지하기 위해 우선적 동일 CAN 네트워크상에서 접속 노드에 대한 인증을 수행한다. 이를 위해, 하나의 MSU를 추가하여 모든 노드가 접속을 완료했을 때 CAN 통신을 시작하는 신호를 지령하게 설계하였다. 모든 노드가 접속하기 전 통신을 시작하는 경우 침입 탐지 알고리즘 실행 여부를 알 수 없으므로 소프트웨어적인 시작 신호를 CAN 통신 동작 동기화를 위해 사용하였다. 각 노드가 네트워크상에서 연결되면 프로세스 초기화를 완료한 후 노드 인증 프로세스를 수행한다. CAN 통신은 브로드캐스팅 통신 방식으로 서버/클라이언트 개념이 없으나, 초기 노드 인증 시에는 1:1 통신을 수행하는 것처럼 동작하도록 알고리즘을 구성하였다.

그림 2는 CAN 네트워크 침입 탐지 검출을 위한 노드의 프로세스 초기화 및 MSU 인증 플로우 차트이다. Ⓐ와 Ⓑ 사이의 수행을 정의하는 접속 인증 프로세스는 그림 3과 같이 도식화할 수 있다.

새롭게 접속한 노드는 MSU로 접속했음을 알리고, MSU는 해당 노드가 접속했음을 인지하고, 해당 노드로 Ack 신호를 보내서 내부 접속 정보를 갱신한다. MSU는 전체 노드가 접속이 완료된 것을 확인하고 난 후, 침입 탐지 방법을 전체 노드에 알린 후, 기존 CAN 통신 데이터의 송수신을 시작하는 신호를 보낸다.

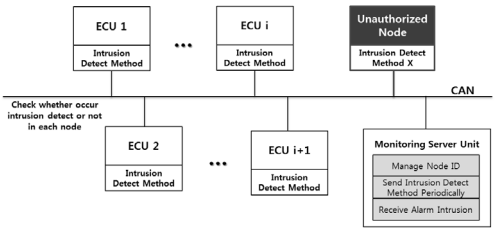

Fig. 1 CAN network intrusion detection system

2.2 전송 노드 침입 탐지 검출 알고리즘

접속 노드가 초기화를 진행하고 외부 침입을 탐지하기 위해 MSU와 접속 노드 인증 프로세스를 완료하면 주어진 주기에 맞도록 송수신 CAN ID 처리 프로세스를 수행한다. 이 때 본 논문에서 제안하는 침입 탐지를 검출할 수 있는 방법이 적용된다. 기 접속한 노드들은 CAN 통신이 시작되기 전 모두 침입 탐지 검출 방법에 대한 정보를 받은 상태이다.

그림 4와 같이 지정된 침입 탐지 검출 방법에 따른 암호화를 수행한다. CAN ID 데이터의 하위 1바이트에 전송 노드 정보를 할당하여 지정된 방법에 따라 노드 정보를 탑재한다. 이때 MSU로부터 침입 탐지 검출 방법으로 전송되는 값은 주기적으로 변경되며, 해당 값에 따른 침입 탐지 검출 방법의 정의 함수는 모든 노드가 알고 있다.

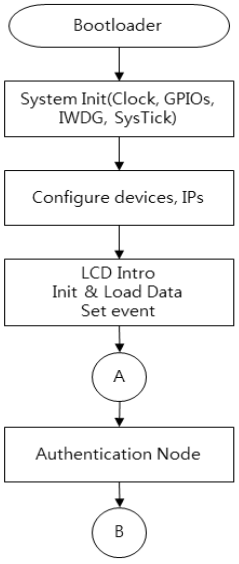

Fig. 2 Flowchart of node authentication

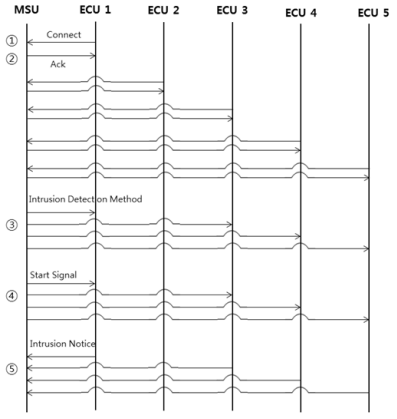

Fig. 3 Process of node connection

MSU에서 발송되는 침입 탐지 검출 방법은 임의로 변경되므로 접속 노드들은 다음에 적용될 침입 탐지 검출 방법에 대해서는 모른다는 것이 핵심이고, 간단하지만 자주, 그리고 쉬운 방법으로 변환된 전송 노드 정보를 통해 외부 침입을 탐지할 수 있도록 설계하였다.

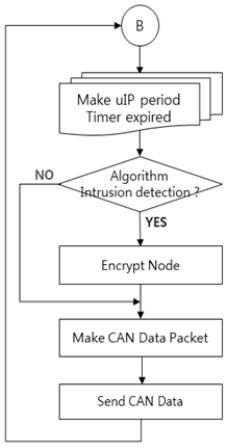

Fig. 4 Flowchart of CAN data transmission process

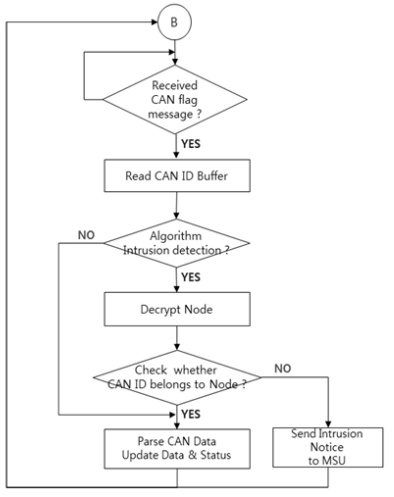

Fig. 5 Flowchart of CAN data reception process

각 노드는 그림 5와 같은 프로세스로 CAN 수신 데이터를 수신 처리한다. 침입 탐지 방법에 대한 정보를 모두 알고 있는 각 노드는 정의된 하위 1바이트 정보를 침입 탐지 검출 방법에 따라 복호화하여 받은 CAN ID의 전송 노드인지를 확인하여 해당 CAN ID를 통한 외부 침입 여부를 판정한다. 이때 침입이 발생하면 MSU로 알림 메시지를 보내 차량 차원의 대응이 가능하도록 조치한다.

각 노드에서 시행하도록 설계된 전송 노드 분석은 네트워크상에서 발생할 수 있는 연산 부하를 줄여주며, MSU로 침입 탐지 알림을 전송하도록 설계한 것은 여러 가지로 발생할 수 있는 외부 침입에 대응하는 차량 동작을 집중해서 처리할 수 있도록 하기 위함이다. 이 모든 처리를 수행함에 있어 다른 CAN ID 들 보다 우선 순위가 높은 ID 값을 사용하여 차량 내 긴급 상황에 대처할 수 있는 통신 알고리즘으로 확장하여 사용 가능도록 설계하였다.

3. 침입탐지 알고리즘 구현 및 성능평가

3.1 침입탐지 시스템 테스트베드 구조

본 논문에서 제안하는 알고리즘을 구현하기 위해서 그림 6과 같이 시스템 하드웨어를 구성하고자 한다. 하드웨어 시스템은 동일 네트워크로 구성되고, MSU에는 차량 속도 모니터링을 위한 LCD와 침입 탐지 알림을 위한 LED가 그림 7과 같이 연결되어있으며, ①초기화 상태, ② Gear D 상태이며 침입 없음, ③ Gear D 상태가 아니며 침입 발생, ④ Gear D 상태이며 침입 발생을 나타낸다.

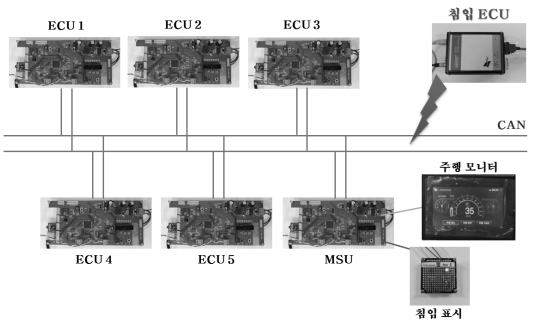

Fig. 6 Test-bed configuration

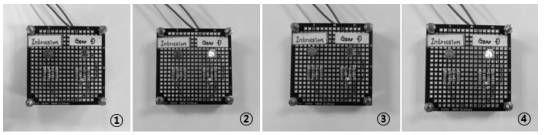

Fig. 7 Intrusion detection notification LED

제안된 알고리즘을 검증하기 위해 CANoe를 이용하여 ECU를 공격하는 침입 시나리오를 그림 8과 같이 구성하였다.

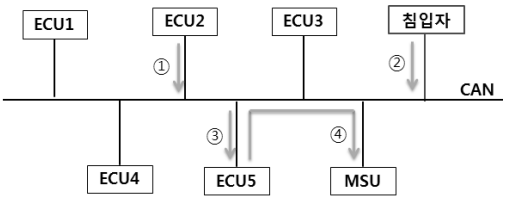

Fig. 8 Intrusion detection simulation

외부 침입자는 기본적으로 CAN ID의 데이터 정보를 알고 있다. 그러나 본 논문에서 제안하는 초기 노드 인증과 전송 노드 침입 탐지 방법에 대한 알고리즘을 모르는 상태이다. 그 상태에서 다음과 같이 침입을 시도하는 시뮬레이션을 구성하였다.

① ECU2에서 침입 탐지 알고리즘을 적용하여 데이터 송신
② ECU2의 CAN 데이터 항목 중 오류 데이터를 넣어 송신 패킷을 만들고 침입
③ ECU2에서 보낸 데이터를 수신하는 ECU5는 침입 탐지 알고리즘이 적용되지 않는 침입자의 CAN 데이터를 수신 받아 침입을 탐지
④ MSU에 침입 탐지를 알림

3.2 시나리오 설계 및 성능평가

본 논문에서 제안하는 CAN 네트워크 침입 탐

지 검출 시스템의 성능을 검증하기 위하여, 다음과 같이 침입 탐지 검출 시나리오를 구성하였다.

성능평가 실험을 위해 차량 구동 제어를 담당하는 VCU(Vehicle Control Unit)의 CAN ID 0x200 내 탑재 데이터인 차량 속력 값과 Shift Pos 로 표시된 기어 위치 값을 외부 침입을 통해 변경하도록 한다. 침입탐지 검출을 위한 데이터 정의는 그림 9와 같다.

CAN ID	Period (ms)	BYTE	BIT	To	Data Name	Min	Max	Res	Unit	Description
0x200	100	0	0	ALL	Node					
		1	8		CR_Brk_RegenL_Nm	-32768	32767	1	rpm	Regen Brake Frce (Low)
		2	16		CR_Brk_RegenH_Nm	-32768	32767	1	rpm	Regen Brake Frce (High)
		3	24		Reserved					
		4	32		Reserved					
		5	40		VCU_Ready	0	1			VCU Ready Signal 0:Not Ready 1:Ready
			41		CF_Veh_Rdy	0	1			Vehicle Ready Signal 0:Not Ready 1:Ready
			42		Regen Ena	0	1			Regen Enable 0:Disable 1:Enable
			43		CF_Tcu_TarGe					Gear Speed (0) P (1) 1 (2) 2 (3) 3 (4) 4 (5) 5 (6) 6 (7) R
			44							
			45							
			46-47		Reserved					
		6	48		CR_ShiftPos					Gear Position (0) P (1) 1 (2) 2 (3) 3 (4) NotUsed (5) D (6) N (7) R (8) Sport (9)-(13) NotUsed (14) IP
			49							
			50							
			51							
			52-55		Reserved					
		7	56		CR_VehSpd_Kph	0	255	1	kph	Vehicle Speed

Fig. 9 CAN ID 0x200 data definition

성능평가를 위한 시나리오는 일정하게 가속되며 운행하던 차량에 외부 침입이 발생하는 경우로, 차량 속도 데이터 오류 발생을 통한 제안 알고리즘의 유효성을 검증하고자 한다.

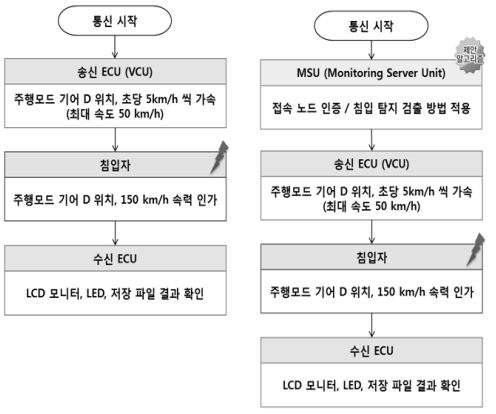

Fig. 10 Scenario of intrusion detection

그림 10과 같이 일정하게 가속하여 운행하는 차량에 과한 속력을 지령하는 외부 침입이 발생했을 때, 알고리즘 비적용 시스템과 알고리즘 적용 시스템의 침입 탐지를 시뮬레이션 하여 결과를 분석한다.

① 시뮬레이션 네트워크에 접속한 모든 노드가 MSU(Monitoring Server Unit) 와 접속 인증을 완료한 후 MSU는 통신 시작 신호를 전송하여 전체 ECU들이 CAN 통신을 시작

② VCU에서 CAN ID 0x200의 기어 위치는 주행 모드 Gear D, 차량 속도는 1초에 5 km/h 씩 증가시켜 CAN 데이터 송신

③ 차속을 제어하는 ECU는 CAN ID 0x200를 수신하여 차속을 반영

④ CAN 데이터 통신 시험 시작 이후 약 4초 이후 CAN ID 0x200에 기어 위치는 주행 모드 Gear D, 차속 150 km/h 정보를 송신하는 외부 침입을 시뮬레이션 하여 인가

⑤ 외부 침입이 탐지되지 않아 과한 속력이 인가되는 것을 확인

⑥ 제안 알고리즘을 적용하여 VCU에서 CAN ID 0x200의 기어 위치는 주행 모드 Gear D, 차속을 1초에 5 km/h 씩 증가시켜 CAN 데이터 송신

⑦ 차속을 제어하는 ECU 또한 제안 알고리즘을 적용하여 CAN ID 0x200를 수신 정보를 분석하여 차속을 반영

⑧ CAN 데이터 통신 시험 시작 이후 약 4초 이후 CAN ID 0x200에 기어 위치는 주행 모드 Gear D, 차속 150 km/h 정보를 송신하는 외부 침입을 시뮬레이션 하여 인가

⑨ 제안 알고리즘 적용으로 외부 침입을 탐지하여 침입자의 데이터는 반영되지 않고, MSU에 외부 침입을 알림

본 논문에서 제안하는 알고리즘을 성능평가하기 위하여 그림 11과 같이 테스트베드를 구축하고, CANoe Trace를 이용해 CAN 네트워크 데이터를 모니터링 하였다.

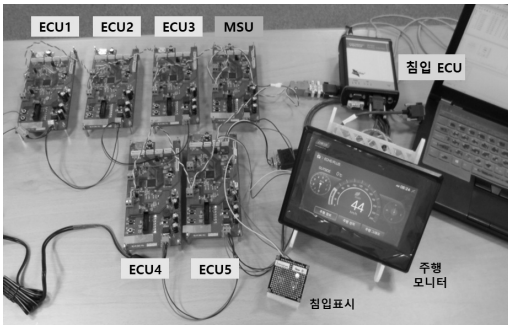

Fig. 11 Intrusion detection performance evaluation test-bed

	LSB							MSB
시속 100 km/h 차속 지령 VCU → ALL	20	DA	0C	00	00	33	05	64
Byte No.	0	1	2	3	4	5	6	7

Fig. 12 Example of 100 km/h speed command

전송되는 데이터 포맷은 그림 12와 같으며, 각 바이트는 다음과 같은 의미를 갖는다. VCU 노드 ID는 0x20이며, CR_Brk_Regen 회생제동 값을 3290 Nm로 임의 값을 입력하면 2바이트에 나누어 표시되면 하위 바이트가 먼저 표시 되므로 0x0CDA 값이 1 번째 바이트에 0xDA, 2번째 바이트에 0x0C 로 할당된다. 5번째 바이트에는 0x33 값이 예시되어 있는데 이는 VCU 수행 준비 상태(1), 차량 Ready 수행 준비(1), Regen Enable 회생제동 안되는 상태(0), 차량 내부 기어 지령 값(6) 이 설정된 것으로, 본 검증에서는 Don't care 정보로 취급한다. 6번째 바이트는 지령 기어 위치로 0x05 값은 주행 모드인 Gear D 로 설정된 값이다. 마지막으로 7번째 바이트의 0x64 값의 10진수 값을 100으로 시속 100km/h 지령 값을 뜻한다.

위에서 설명한 Shift Pos 즉 기어 위치 지령 값과 차량 속도 값을 통해 제안 알고리즘의 외부 침입 탐지 적정성을 확인할 수 있다.

그림 13.(a)는 일반적인 CAN 데이터 전송 패킷을 나타내는 CAN ID 데이터 모니터링 화면으로 왼쪽부터 하위 바이트로 정의된다. 0x10 에서부터 0x40으로 정의된 4개의 모듈에서 전송되는 데이터가 표시되었다. 노드 ID 값에서 확인된 바와 같이 제안 알고리즘이 적용되지 않은 것을 확인할

190,636002	CAN 1 502	Rx	8	40	5E	01	00	00	00	00	00
190,636244	CAN 1 501	Rx	8	40	C4	09	C4	09	00	00	00
190,636495	CAN 1 670	Rx	8	10	2C	01	5E	02	00	00	00
190,636739	CAN 1 671	Rx	8	10	10	10	7D	F0	D8	E0	0D
190,636987	CAN 1 620	Rx	8	10	00	00	00	00	3C	D3	78
190,637231	CAN 1 623	Rx	8	30	B8	0B	B8	0B	00	00	01
190,637491	CAN 1 295	Rx	8	30	00	00	00	00	00	00	00
190,637739	CAN 1 611	Rx	8	20	07	02	00	00	00	00	00
190,637983	CAN 1 200	Rx	8	20	DA	0C	00	00	33	05	27

(a) CAN data transmission per connected node

14,336664	CAN 1 502	Rx	8	40	5E	01	00	00	00	00	00
14,336907	CAN 1 501	Rx	8	40	C4	09	C4	09	00	00	00
14,337159	CAN 1 670	Rx	8	10	2C	01	5E	02	00	00	00
14,337401	CAN 1 671	Rx	8	10	10	10	7D	F0	D8	E0	0D
14,337651	CAN 1 620	Rx	8	10	00	00	00	00	3C	D3	78
14,337893	CAN 1 623	Rx	8	30	B8	0B	B8	0B	00	00	01
14,338153	CAN 1 295	Rx	8	30	00	00	00	00	00	00	00
14,338401	CAN 1 611	Rx	8	20	07	02	00	00	00	00	00
14,338645	CAN 1 200	Rx	8	20	DA	0C	00	00	33	05	27
14,164146	CAN 1 200	Tx	8	20	DA	0C	00	00	33	05	96

(b) External intrusion

Fig. 13 Not using algorithm

1007,090625	CAN 1 502	Rx	8	10	5E	01	00	00	00	00	00
1007,090863	CAN 1 501	Rx	8	10	C4	09	C4	09	00	00	00
1007,091115	CAN 1 670	Rx	8	04	2C	01	5E	02	00	00	00
1007,091357	CAN 1 671	Rx	8	04	10	10	7D	F0	D8	E0	0D
1007,091610	CAN 1 620	Rx	8	04	00	00	00	00	3C	D3	78
1007,091852	CAN 1 623	Rx	8	0C	B8	0B	B8	0B	00	00	01
1007,092108	CAN 1 295	Rx	8	0C	00	00	00	00	00	00	00
1007,092354	CAN 1 611	Rx	8	08	07	02	00	00	00	00	00
1007,092598	CAN 1 200	Rx	8	08	DA	0C	00	00	33	05	0E

(a) CAN data transmission per connected node

1065,087908	CAN 1 502	Rx	8	10	5E	01	00	00	00	00	00
1065,088146	CAN 1 501	Rx	8	10	C4	09	C4	09	00	00	00
1065,088396	CAN 1 670	Rx	8	04	2C	01	5E	02	00	00	00
1065,088640	CAN 1 671	Rx	8	04	10	10	7D	F0	D8	E0	0D
1065,088892	CAN 1 620	Rx	8	04	00	00	00	00	3C	D3	78
1065,089134	CAN 1 623	Rx	8	0C	B8	0B	B8	0B	00	00	01
1065,089390	CAN 1 295	Rx	8	0C	00	00	00	00	00	00	00
1065,089636	CAN 1 611	Rx	8	08	07	02	00	00	00	00	00
1065,089880	CAN 1 200	Rx	8	08	DA	0C	00	00	33	05	1E
1065,111487	CAN 1 200	Tx	8	20	DA	0C	00	00	33	05	96

(b) External intrusion detection

Fig. 14 Using algorithm

수 있다. 현재 0x200 CAN ID 데이터를 보면, 차량 속도는 0x27 즉 39 km/h, 주행 모드인 Gear D 위치 값인 0x05 값으로 지령하고 있다. 데이터 확인을 위해 차량 속도는 1초에 5km/h 씩 상승하도록 하였다.

그림 13.(b)는 외부침입 CAN 데이터 전송을 나타내고 있다. 이때 마지막 줄과 같이 0x200 CAN ID 데이터 형식으로 구성된 외부 침입이 발생하였다. 차량 속도 지령 값이 0x96 즉, 150km/h, 주행 모드인 Gear D 위치인 0x05 값을 확인할 수 있다. 동일 네트워크상에서 0x200 CAN ID 값을 사용하는 모듈은 외부 침입을 인지하지 못하고 갑자기 들어온 고속의 정보를 반영할 수 있다. 이러한 경우, 시속 100 km/h 이상 값이 차량에 반영되는 경우 급발진과 같은 동작으로 도로 상에서는 큰 사고로 이어질 수 있다.

그림 14는 본 논문에서 제안하는 알고리즘을 적용한 경우이며, 노드 ID 값이 원래의 것과 다른 것을 확인할 수 있다. 침입 탐지 검출 방법이 적용된 값이며, 노드 ID 값 모두 오른쪽으로 2회 shift 연산을 한 것을 확인할 수 있다.

현재 0x200 CAN ID 데이터를 보면, 차량 속도는 0x0E 즉 14 km/h, 주행 모드인 Gear D 위치 값인 0x05 값으로 지령하고 있다. 데이터 확인을 위해 차량 속도는 1초에 5km/h 씩 상승하도록 하였다.

이때 그림 14(b)의 마지막 줄과 같이 0x200 CAN ID 데이터 형식으로 구성된 외부 침입이 발생하였다. 차량 속도 지령 값이 0x96 즉, 150km/h, 주행 모드인 Gear D 위치인 0x05 값을 확인할 수 있다. 동일 네트워크상에서 인증과정을 수행한 모듈들은 침입 탐지 검출 방법에 대해서 공유하고 있기 때문에 외부 침입을 탐지할 수 있고, MSU로 침입을 통지할 수 있다. 차량 동작 오류를 발생시킬 수 있는 데이터를 원천적으로 봉쇄함으로써 사고를 방지할 수 있음을 확인할 수 있다.

본 논문에서 제안하고 있는 CAN 네트워크 침입 탐지 검출 성능평가 결과 그래프를 그림 15로 나타내었다. 상위 2개의 파란색 결과 그래프는 제안 알고리즘 미적용 시스템을 평가한 것으로서 급작스런 CAN 네트워크 침입으로 인해 주행 중 차량 속도가 150 km/h 로 급격히 변동되는 것을 확인할 수 있다.

침입탐지 알고리즘이 적용된 아래 2개의 빨간색 그래프는 차량 오류를 발생시킬 수 있는 데이터의 원천 봉쇄하여 오류 신호가 입력되지 않음을 확인할 수 있었다.

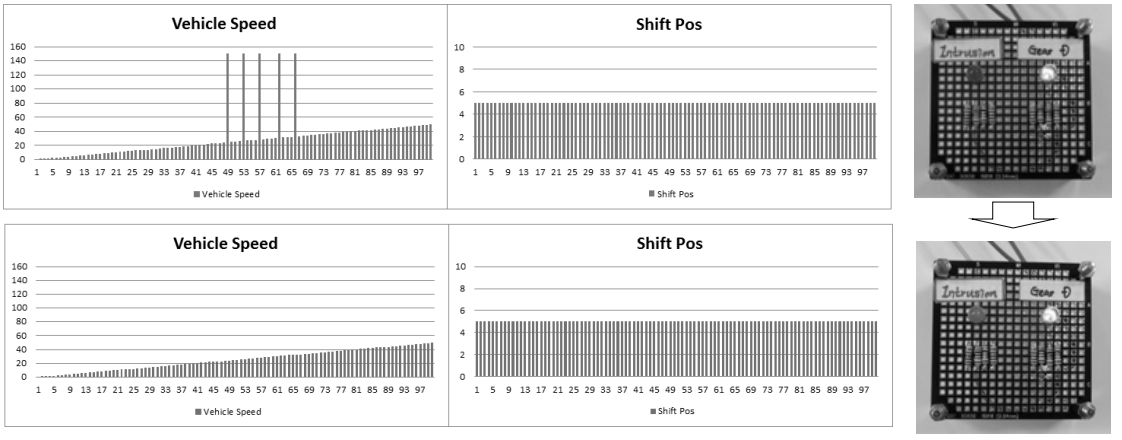

Fig. 15 Results of Intrusion Landing Performance Evaluation

4. 결론

본 논문에서는 커넥티드 카에서 보안의 취약점이 될 수 있는 CAN 네트워크의 침입 탐지를 검출하기 위한 방법을 제안하였다.

제안 알고리즘은 단위 네트워크에 모니터링 서버 노드를 추가 장착하고, 접속하는 노드의 확인하고, 해당 네트워크 내에 노드 암호화 방법을 주기적으로 달리하여 CAN 데이터 패킷을 생성하도록 하는 것이다. 모니터링 서버 노드는 최초 인증과 노드 암호화 방법 전송, 그리고 접속 노드에서 발견되는 침입을 받아 조치를 취하는 기능을 수행한다. 각 접속 노드는 모니터링 서버 노드로부터 전달받은 침입 탐지 검출 방법에 따라 암호화/복호화를 수행하는데, 수신 CAN ID의 하위 바이트에 배치된 노드 값을 침입 탐지 검출 방법에 따라 복호화 하여 알고 있는 노드 정보와 CAN ID가 서로 다르면 외부 침입으로 간주하고 모니터링 서버 노드로 침입이 탐지되었음을 알린다. 이와 같은 과정을 통해 외부에서 유입되는 오류 데이터에 대한 원천 차단이 가능하고, 침입을 인지한 모니터링 서버 노드는 차량 시스템 보안을 위한 협력 시스템 구축이 가능해진다.

현재까지 제안된 내부 네트워크 보안 취약점을 개선하기 위한 방법은 복잡한 인증 알고리즘을 구현하여 탑재함으로써 한정적인 CAN 데이터 영역을 많이 할당해야하고, 연산이 복잡한 알고리즘을 이용한 CAN 데이터 처리는 고기능, 고사양을 요구하는 시스템 부하 및 통신 부하의 원인이 되기도 했다.

하지만 제안하는 알고리즘의 경우 단위 네트워크 내의 접속하는 노드의 인지와 암호화 방법 브로드캐스팅, 외부 침입 탐지 알림에 대한 수신만 수행하도록 설정하고 있으므로 간단한 하드웨어 구성만으로 기능을 구현할 수 있기 때문에 추가되는 노드라는 부담은 훨신 적어질 수 있다.

그리고 CAN 데이터에 포함되는 전송 노드 정보는 1 바이트 ID를 부여하여 단순화시켰으며, 암호화/복호화 방법 간단한 shift 연산, 더하기/빼기 연산 등으로 구성하여 침입 탐지 알고리즘이 추가되었음에도 연산부하와 통신 부하를 최소한으로 줄이기 위해 노력하였다. 또한, 외부 노출 위험을 보완하기 위해 주기적으로 침입 탐지 검출 방법을 업데이트해 주었다. 제안 알고리즘의 유효성은 시뮬레이션 성능평가를 통해 검증하였다.

향후 연구에서는 외부 통신이 다양하게 연동되어 있는 실제 커넥티드 카의 환경을 구축하여 시뮬레이션을 수행하고, 최종적으로 직접적인 외부 노이즈나 침입이 존재하는 실제 커넥티드 카에서의 성능 실험을 통해 유효성을 검증하는 것이 필요하다.

후기

이 논문은 부경대학교 자율창의학술연구비(2016년)에 의하여 연구되었음.

참고문헌

[1] 조아람, “CAN 버스 공격에 안전한 메시지 인증 및 키 분배 메커니즘”, 석사학위논문, 고려대학교 (2013).
[2] “자능형 교통시스템의 차량 통신 보안 기술 동향과 전망”, 한국방송통신전파진흥원, 제59호, (2014).
[3] 이혜련, 김경진, 정기현, 최경희, 박승규, 권도근, “자동차용 ECU의 CAN 메시지를 통한 자동차 공격 방법 연구”, 한국컴퓨터정보학회논문지, 18(11), pp.39-49, (2013).
[4] 명의정, 윤주범, “커넥티드카 보안 위협 동향 연

구", 한국통신학회, (2015).
[5] 문보경, "[이슈분석]커넥티드카 '합종연횡'..수직계열화 지고 융합·협력이 대세", 전자신문, (2016.05.16.).
[6] 김선영, "커넥티드 카 서비스 동향 분석", 주간기술동향, (2013).

(접수:2017.04.05. 수정: 2017.05.02. 게재확정: 2017.05.12.)