정 보 보 호 학 회 지
제 33 권 제4호, 2023. 8

CAN 네트워크에서의 악의적인 ECU 식별 기술 연구 동향

이 세 영*, 최 원 석**, 이 동 훈***

요 약

자동차 산업에서 전자제어장치 (Electronic Controller Unit, ECU)를 활용한 혁신으로 운전자들은 안전하고 편리한 운전 경험을 누리고 있다. 그러나 이와 동시에, 차량 내부 ECU 간의 통신을 지원하는 CAN (Controller Area Network)을 대상으로 한 악의적인 침입과 사이버 공격의 위협 역시 증가하고 있다. 이러한 문제에 대응하기 위해 많은 연구가 진행 중이며, 특히 자동차 침입 탐지 시스템 (Intrusion Detection System, IDS)의 발전이 주목받고 있다. 그러나 대부분의 IDS는 주로 공격을 탐지하는 데 집중되어 있으며, 실제 악의적인 메시지를 전송한 ECU를 정확히 식별하는 데에는 한계점이 있다. 악의적인 ECU를 식별하는 기술은 공격 ECU를 격리시키거나 펌웨어 업데이트 등의 보안 패치를 적용하는 데 필수적인 기술이다. 본 고에서는 현재까지 제안된 CAN에서의 악의적인 ECU를 식별하기 위한 기술들에 대해 살펴보고, 비교 분석 및 한계점에 대해 분석하고자 한다.

Ⅰ. 서 론

CAN (Controller Area Network) 프로토콜은 1980년대 Bosch 사에서 개발된 차량 내부 네트워크로서 Bus 형태의 네트워크 토폴로지를 지원한다[1]. CAN은 비용적인 효율성과 높은 통신 신뢰성을 제공하며, 특히 외부 노이즈에 견고한 통신 환경을 제공한다. 현대의 최신 차량에는 수십 개 이상의 전자제어장치 (Electronic Controller Unit, ECU)가 탑재되어 있으며, 각 ECU는 차량의 다양한 시스템을 관리하고 제어한다. 예를 들어 엔진 제어, 브레이크 제어, 안전 시스템 등 각각의 기능을 각 ECU가 담당한다. 이러한 ECU들은 CAN Bus를 통해 실시간으로 데이터를 주고받으며, 차량의 운전과 안전에 중요한 역할을 수행한다.

그러나 CAN 프로토콜에는 암호화나 인증과 같은 보안 메커니즘의 적용되어 있지 않아 외부에서 CAN Bus에 접근한 공격자는 이를 악용하여 악의적인 메시지를 전송할 수 있다[2]. 이로 인해 차량에 오동작이 발생할 수 있으며, 이는 심각한 사고를 초래할 수도 있다. 실제로 2015년 C.Miller와 C.Valasek에 의해 원격으로 Jeep Cherokee 차량을 해킹하여 디스플레이, 브레이크, 핸들 제어 등 공격자가 임의로 차량을 조작할 수 있음이 발표되었다[3]. 더 최근에는 Tencent의 Keen security lab에서 테슬라 차량의 취약점을 통해 차량 제어가 가능함이 발표되었다[4].

이러한 보안 취약점을 보완하기 위해 여러 연구들이 진행되어 왔다. 초기 CAN 프로토콜에 암호화나 인증과 같은 보안 메커니즘을 적용하는 연구[5,6]들이 진행되었지만, 실제 차량 환경에서 보안 메커니즘을 적용하는 것에는 다양한 한계점이 있다. CAN 프로토콜은 실시간 응답성이 중요한 환경이므로 보안 기능 추가로 인해 데이터 전송 지연이 발생할 수 있으며, 이는 실시간 성능에 저하를 일으킨다. 또한, CAN은 제한된 데이터 Payload를 가지고 있어 보안 데이터를 주고 받는 것에 제한이 있다.

실질적으로 CAN 프로토콜에 암호화나 인증과 같은 보안 메커니즘을 적용하는 것에 한계가 있음에 따라, CAN Bus에 주입되는 악의적인 CAN 메시지를 탐지하는 IDS (Intrusion Detection System) 연구가 최근 각광을 받고 있다[7-9]. IDS는 CAN Bus를 모니터링하여 공격자에 의해 주입된 공격 메시지를 탐지

본 연구는 과학기술정보통신부의 2021년 자율주행기술개발혁신사업인 '자율주행차량의 차세대 내부 네트워크의 보안 및 초고속 무결성 부여 기술 개발'(No. 2021-0-01348)의 지원을 받았습니다.
* 고려대학교 정보보호대학원 (대학원생, seyoung0131@korea.ac.kr)
** 고려대학교 정보보호대학원 (조교수, beb0396@korea.ac.kr)
*** 고려대학교 정보보호대학원 (교수, donghlee@korea.ac.kr)

하며, 이로 인해 기존 차량 시스템 및 ECU의 변경을 요구하지 않고 차량에 원활하게 통합할 수 있다. 실제로 자동차 IDS 제품에 대한 연구 및 개발은 이제 학술적인 영역을 넘어서서 진행되고 있다. 미시간 대학교 교통연구소 (University of Michigan Transportation Research Institute, UMTRI) 는 자동차 제조사들의 펀딩을 받아 자동차 IDS 제품들의 성능을 평가하는 방법에 대한 프로젝트를 수행하였으며, 이는 가까운 미래에 자동차 제조사들이 자동차 IDS를 차량에 설치할 계획임을 시사한다[10].

최근까지 다양한 IDS 기술이 제안되어 왔으나, IDS는 주로 공격을 탐지하는 데 집중되어 있으며, 공격을 탐지한 이후 이에 대응하는 것에 대한 연구는 미비한 상태이다. 특히, CAN 메시지에는 해당 메시지를 어떤 ECU가 전송한 것인지에 대한 식별 정보가 포함되어 있지 않기 때문에, ECU를 식별하는 연구는 공격 탐지 이후 대응을 위해서 반드시 필요한 기술이다. 공격 CAN 메시지를 전송한 악의적인 ECU를 식별하는 기술을 통해 악의적인 ECU를 CAN Bus로부터 격리시키거나 위변조된 ECU를 다시 정상 펌웨어로 업데이트하는 등의 공격 탐지 이후 대응이 가능하다. 본 고에서는 현재까지 제안된 CAN에서의 ECU 식별 기술을 살펴보고 각 기술들을 비교하고 이들의 한계점에 대해 분석하고자 한다.

본 고의 구성은 다음과 같다. 2장에서는 CAN 네트워크와 관련된 배경지식을 소개하고, 3장에서는 현재까지 연구된 ECU 식별 기술들에 대해 분석한다. 4장에서는 각 기법들의 한계점들에 대해 설명하며 비교 분석한다. 마지막으로 5장에서는 결론을 맺는다.

Ⅱ. CAN 네트워크 배경지식

2.1. ECU

자동차 분야에서 CAN Bus에 연결된 장치들을 일반적으로 노드라 지칭하며, 특히 ECU라고 알려져 있다. ECU는 일반적으로 MCU (Microcontroller Unit), CAN Controller, CAN Transceiver 세 가지 주요 구성 요소로 구성되어 있다. 일반적으로 공격자는 다양한 취약점을 바탕으로 MCU의 소프트웨어를 악의적으로 위변조하여 CAN Bus에 대한 공격을 수행한다. MCU와 달리 CAN Controller와 CAN Transceiver는 하드웨어 레벨에서 구현되어 동작하기에 공격자에 의해 원격에서 위변조되기 어렵다는 특징이 있다.

2.1.1. MCU

MCU는 프로그래밍 가능한 프로세서로서, 자동차 시스템의 작동을 어플리케이션 (Application) 계층에서 제어하고 조정하는 데 사용된다. 다양한 센서에서 수집한 데이터를 처리하고, 주행 상태를 모니터링하며, 다른 ECU로부터 전송받은 데이터를 처리하여 여러 가지 제어 기능을 실행한다.

2.1.2. CAN Controller

CAN Controller는 데이터 링크 (Data Link) 계층에서 작동하는 구성 요소로서, MCU와 CAN 버스의 물리적 (Physical) 계층 사이를 연결하고 중개한다. CAN Controller에서 데이터를 프레임으로 구성하여 다른 ECU들과의 효율적인 통신을 지원하며, 통신 중에 발생한 다양한 오류에 대해 처리를 수행한다.

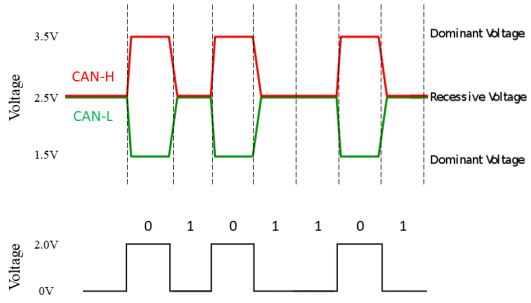
2.1.3. CAN Transceiver

CAN Transceiver는 물리적 계층에서 동작하는 구성 요소로서, CAN Controller가 생성한 프레임 비트 스트림을 아날로그 Voltage로 변환하여 CAN Bus에 송신하고, 수신된 아날로그 Voltage를 다시 디지털 비트 스트림으로 변환하여 CAN Controller를 통해 MCU가 이해할 수 있도록 한다.

2.2. CAN 네트워크

2.2.1. Signal Level and Bit Representation

CAN Bus 네트워크는 CAN High (CAN-H)와 CAN Low (CAN-L)로 불리는 두 개의 전기적인 라인으로 구성된다. 외부의 전자기 간섭과 노이드에 대한 견고성을 보장하기 위해 CAN 프로토콜은 CAN-H와 CAN-L 사이의 Voltage Level 차이를 이용하여 개별 비트를 Dominant bit (0) 또는 Recessive bit (1)으로 나타낸다. ISO 11898-2에서 정의한 High-speed CAN의 경우 두 전기 선의 Voltage Level 차이가 0.5 V 미만일 경우 bit 1로 정의하고, Voltage Level 차이가 0.9 V를 초과하면 bit 0으로 정의한다. [그림 1]은

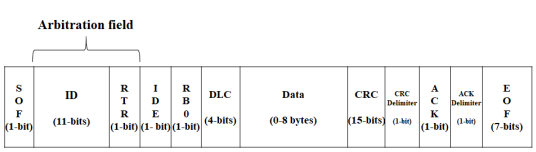


〔그림 1〕 High-speed CAN에서의 CAN-H와 CAN-L의 Voltage 신호 예

High-speed CAN에서 Dominant bit (0)과 Recessive bit (1)을 전송할 때의 CAN-H와 CAN-L의 Voltage Level을 보여준다. 만약 Dominant bit (0)과 Recessive bit (1)이 CAN Bus에 동시에 전송되는 경우 Voltage 차이가 큰 dominant bit (0)이 CAN Bus를 점유한다.

2.2.2. CAN 데이터 프레임과 Arbitration

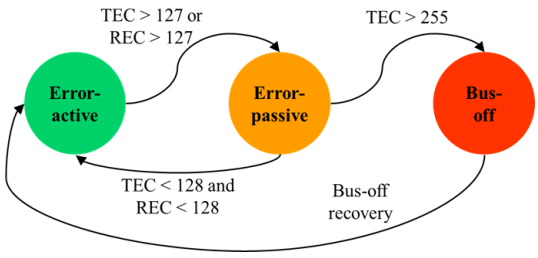
CAN의 통신을 위해 정의된 프레임 중 ECU들의 데이터 송·수신을 위해 가장 많이 사용되는 것은 데이터 프레임이다. CAN 2.0 프로토콜은 CAN ID의 길이에 따라 11 bits ID를 가지는 CAN 2.0 A (Standard Format), 29 bits의 확장된 ID를 가지는 CAN 2.0 B (Extended Format)로 나누어 정의된다. 이 중 가장 일반적으로 사용되는 데이터 프레임은 11 bits ID를 가지는 데이터 프레임이며, 프레임 구조는 [그림 2]와 같다. SOF (Start-of-Frame) 필드 (Field)는 데이터 프레임의 시작을 나타내는 필드로 1-bit 영역이 할당되어 있으며, 항상 dominant bit (0)으로 정의되어 있다. ID 필드는 데이터 프레임의 식별자 (Identifier 또는 CAN ID) 정보를 나타낸다. RTR (Remote Transmission Request) 필드는 데이터 프레임 또는 리모트 프레임을 구분하는 영역으로 1-bit가 할당되어 있으며, 모든 데이터 프레임은 이 영역이 항상 dominant bit (0)로 정의되어 있다. 하나의 데이터 프레임에는 최대 8 bytes의 데이터를 전송할 수 있다. 본 고에서는 편의를 위해 데이터 프레임을 메시지라 명명한다. CAN Bus가 유휴 (Idle)인 경우 모든 ECU는 CAN 메시지 전송을 시도할 수 있으며, 동시에 두 개 이상의 ECU가 CAN Bus에 메시지를 전송하는 경우 Arbitration Filed 값을 바탕으로 전송 우선순위를 결정한다. Arbitration 과정은 데이터 프레임 내 Arbitration Field에서 이루어지며 ID Field 값이 작을수록 메시지 전송 우선순위가 높다. Arbitration 과정에서 이긴 ECU만이 데이터 프레임 전송을 지속하며, Arbitration 과정에서 밀린 ECU는 즉시 데이터 프레임 전송을 중지하고 다음 CAN Bus 유휴 상태에 다시 데이터 프레임 전송을 시도한다.



〔그림 2〕 CAN 데이터 프레임 (Standard Format)

2.2.3. Error handling과 Fault confinement

CAN 프로토콜에는 CAN 통신 중 발생하는 Error에 대해 대응하기 위해 Error Handling과 Fault confinement 메커니즘이 정의되어 있다. CAN에 발생 가능한 Error는 총 5가지가 있으며, 그중에 Bit Error는 Arbitration Field 이후 현재 CAN Bus에 메시지를 전송 중인 ECU가 자신이 보낸 Bit와 다른 Bit가 CAN Bus에 감지될 경우 발생하는 Error이다. 각 ECU에는 TEC (Transmit Error Counter)와 REC (Receiver Error Counter)라는 카운터가 정의되어 있으며, ECU가 메시지를 전송 중일 때 Error가 발생되는 경우 TEC를 증가시키고 ECU가 메시지를 수신 중일 때 Error 가 발생되는 경우 REC를 증가시킨다. 특히, ECU가 메시지를 전송 중일 때 Error를 인지하면 해당 ECU의 TEC는 8 증가한다. 두 Error 카운터에 의해 [그림 3]과 같이 ECU의 Error 상태가 결정된다. Error Active 상태는 ECU의 초기 상태 (default)로



〔그림 3〕 TEC와 REC에 따른 ECU의 Error 상태

TEC와 REC 각각 0으로 시작한다. Error Active 상태인 ECU는 통신 중 Error 인지 시 Active Error Flag (000000)을 전송하여 다른 ECU에 Error를 알린다. 지속적인 Error로 TEC 또는 REC가 127을 초과하면 Error Passive 상태로 전환된다. Error Passive 상태인 ECU가 Error를 인지하는 경우 Passive Error Flag (111111)을 전송하게 되고, 이는 CAN Bus에 어떠한 영향도 주지 않는다. 특히, TEC가 255를 초과하면 ECU는 Bus-off 상태로 전환된다. Bus-off 상태인 ECU는 더 이상 CAN 통신을 수행할 수 없다. Bus-off 상태인 ECU는 CAN Bus에서 11개의 연속적인 recessive bits가 128번 모니터링되면 다시 Error Active 상태로 전환되며 CAN 통신을 수행한다 [1].

Ⅲ. CAN에서의 ECU 식별 기법

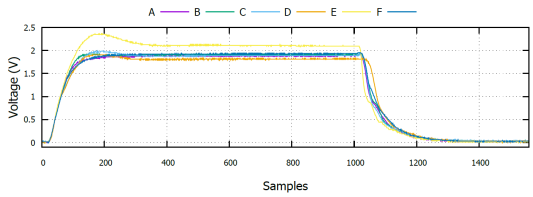
ECU 식별 기술이란 Bus에 전송된 메시지에 대해서 어떤 ECU가 전송하였는지 식별하는 기술이다. ECU마다의 고유한 하드웨어적인 특징을 전송된 메시지에서 추출하여 식별에 활용하거나, 특정 ECU의 Error 상태를 분석함으로써 해당 메시지를 전송한 ECU에 대해서 식별하는 연구가 소개되었다. ECU마다의 하드웨어적인 특징으로는 각 ECU의 CAN Transceiver에서 발생하는 고유한 Voltage 신호의 특징을 기반으로 식별하는 방법과 CAN Controller의 고유한 내부 Clock으로부터 생성되는 Clock-skew를 활용하여 식별하는 기법으로 분류할 수 있다. ECU의 고유한 하드웨어적인 특징을 이용하는 방법 외에도 ECU를 특정 Error 상태로 강제로 천이시키고 실시간으로 이를 확인하여 해당 CAN 메시지를 전송한 ECU를 식별하는 방법이 있다. 본 장에서는 이러한 ECU 식별 기술에 대해 소개한다.

3.1. Voltage 신호 기반 식별 방법

Voltage 신호 기반 식별 방법은, 각 ECU가 동일한 비트 스트림을 CAN Bus에 전송한다 할지라도 CAN Bus에 나타나는 Voltage 신호는 ECU 별로 고유하다는 특징을 활용한다. 이를 위해, ECU 내 고유한 CAN Transceiver에서 생성한 Voltage 신호의 유일성과 CAN Bus에 연결된 ECU의 고유한 위치로 인한 Voltage 신호의 Propagation time을 활용한다.

3.1.1. 통계적인 특징 기반 식별 방법

각 ECU가 동일한 비트 스트림을 CAN Bus를 전송할지라도, CAN Bus에 나타나는 물리적인 Voltage 신호는 각 ECU 내 존재하는 유일한 CAN Transceiver에 의해 서로 다른 특징이 나타난다[12]. [그림 4]는 서로 다른 여섯 개의 ECU가 동일한 메시지를 보낼 때 Voltage 신호 간의 차이가 존재함을 보여준다. Voltage 신호 간의 구별되는 차이를 분석하기 위해 이러한 연구에서는 CAN 데이터 프레임의 Voltage 신호를 수집하고 수집한 Voltage 신호로부터 시간 또는 주파수 도메인에서의 다양한 통계적인 특징 (Statistic features)을 분석한다. 그리고 추출된 특징으로 분류기 (Classifier)를 학습하고 임의의 메시지가 전송되는 경우 해당 메시지를 전송한 ECU에 대해 식별을 수행한다[12-14].



〔그림 4〕 동일한 비트 스트림에 대한 다양한 ECU의 Voltage 신호 차이 〔11〕

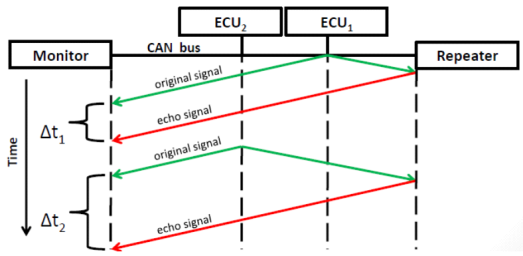
〔표 1〕 Voltage 신호로부터 계산되는 Features 〔13〕

Feature	Description
Mean	$\mu = \frac{1}{N}\sum_{i=1}^{N} x(i)$
Standard Deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x(i)-\mu)^2}$
Variance	$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x(i)-\mu)^2$
Skewness	$skew = \frac{1}{N}\sum_{i=1}^{N}(\frac{x(i)-\mu}{\sigma})^3$
Kurtosis	$kurt = \frac{1}{N}\sum_{i=1}^{N}(\frac{x(i)-\mu}{\sigma})^4$
Root Mean Square	$s = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x(i)^2}$
Maximum	$max = max(x(i))_{i=1\ldots N}$
Energy	$en = \frac{1}{N}\sum_{i=1}^{N} x(i)^2$

차량의 실제 CAN Bus 상에서 Voltage 신호를 실용적으로 수집하기 위해 다양한 연구들이 진행되고 있다. 예를 들어, 전체 CAN 데이터 프레임의 Voltage 신호가 아닌 CAN 데이터 프레임 내 특정 영역의 Voltage 신호만을 사용하여 식별하는 연구가 발표되었다. 이와 별개로 CAN Bus의 Voltage 신호 수집 시, CAN-H와 CAN-L의 Voltage 신호를 별개로 수집하여 식별 정확도를 올리고자 하는 연구가 수행되었다. [표 1]은 Voltage 신호로부터 특징을 추출하기 위해 가장 많이 활용되는 통계적인 항목을 나타낸다.

3.1.2. Propagation 시간 기반 식별 방법

CAN Bus 네트워크는 CAN-H와 CAN-L 두 전기적 라인으로 구성되며, Voltage 신호의 전송에 걸리는 시간은 송신 ECU와 수신 ECU 간의 거리에 영향을 받는다. Voltage 신호의 Propagation 시간을 기반으로 하는 접근 방식은 해당 시간 간격을 기반으로 ECU의 위치를 결정하여 메시지를 전송한 ECU를 식별한다[15, 16]. CAN Bus에 연결된 ECU들은 항상 고정된 위치에서 Voltage 신호를 생성한다. 이러한 Voltage 신호로부터 Propagation 시간을 추출하기 위해 [그림 5]와 같이 Voltage 신호의 측정 장치 (Monitor)와 이를 중계하는 중계기 (Repeater)를 CAN Bus 양 끝에 설치하고, 중계기에서 생성한 에코 신호를 활용하여 특정 메시지를 전송한 ECU의 위치를 계산한다. 측정된 거리에 따라 ECU를 식별하기 위해 각 ECU의 Propagation 시간이 계산된 인증 테이블이 필요하다[11].



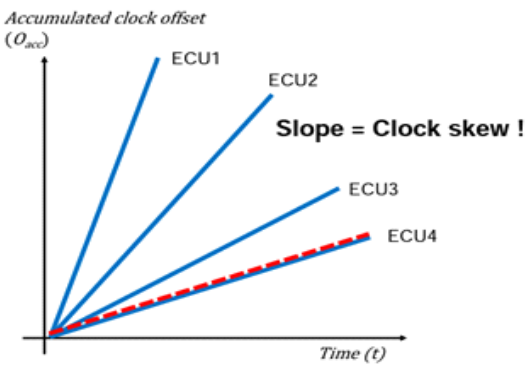
〔그림 5〕 특정 위치에서 전송하는 Voltage 신호에 대한 양 끝단에서의 Propagation 시간 차이 〔15〕

3.2. Clock-skew 기반 식별 방법

Clock-skew 기반 식별 방법은 ECU 내 주기적인 메시지 전송이나 시간 동기화 등을 위해 사용되는 CAN Controller 하드웨어의 Clock-skew를 추출하여 ECU를 식별하는 방법이다. 동일한 시간을 측정하더라도 각 Clock 간의 미세한 제조 공정 차이로 인해 Clock offset이 존재하며, 이러한 차이는 특정 ECU에 대한 Clock-skew 추출을 가능하게 한다. 어떤 시간 정보로부터 Clock-skew를 추출하느냐에 따라 두 가지 방법으로 분류할 수 있다.

3.2.1. 메시지 수신 시간 기반 식별 방법

일반적으로 CAN Bus에 연결된 ECU들은 CAN 메시지를 주기적으로 전송한다. 따라서 주기적으로 수신되는 특정 CAN ID 메시지에 대해 메시지의 수신 간격을 측정하여 Clock-skew를 추출하는 방법이다[17,18]. 정해진 주기에 의해 예상되는 메시지의 수신 시간과 실제 수신 시간 간의 시간 간격을 이용하여 Clock offset을 계산하고, 누적된 Clock offset으로부터 특정 CAN ID 메시지에 대한 Clock-skew를 추출한다. [그림 6]은 서로 다른 ECU 간의 누적된 Clock offset과 이로부터 계산된 Clock-skew를 나타낸다. 특정 CAN ID 메시지가 수신되는 경우 Clock-skew를 계산하고 이전에 알고 있는 Clock-skew 정보와 비교하여 해당 CAN 메시지를 전송한 ECU를 식별한다.



〔그림 6〕 서로 다른 ECU 별 Clock-skew의 차이 〔17〕

3.2.2. 데이터 프레임 내 Bit time 기반 식별 방법

메시지 수신 시간에 기반하여 Clock-skew를 추출하는 방법은 수십 개의 여러 주기적인 CAN 메시지가 수신되어야 정확한 Clock-skew 추출이 가능하다. 해

당 방법은 수십 개의 CAN 메시지가 아니라 단일 CAN 메시지 내에서 전송된 데이터 프레임의 각 Bit timing을 바탕으로 Clock-skew를 추출하는 방법이다 [19,20]. Arbitration 과정 이후에는 CAN Bus 상으로 하나의 ECU만이 메시지 전송이 허용된다. 따라서 CAN Bus에 전송 중인 메시지에 대한 실제 Bit time과 예상되는 Bit time을 비교하여 Clock offset 및 Clock-skew 추출이 가능하다. 실제 CAN Bus에 전송되는 Bit time을 정확히 파악하기 위하여 Voltage 신호 기반 식별 방법과 동일하게 CAN Bus의 Voltage 신호를 수집하여 해당 CAN 메시지의 실제 Bit time에 대해 정확히 산출한다. 이후, 특정 CAN ID 메시지의 Bit time으로부터 Clock-skew를 추출하고 기존 ECU들의 Clock-skew와 비교하여 해당 메시지를 전송한 ECU를 식별한다. 해당 방법은 단일 Bit에 대해서 Bit time을 계산하는 방법과 전체 단일 프레임에 대해서 Bit time을 계산하는 방법으로 나눌 수 있다.

3.3. ECU의 Error 상태 기반 식별 방법

ECU의 Error 상태 기반 방법은, 앞서 설명한 ECU의 고유한 하드웨어 특징을 추출하여 식별하는 연구와는 달리, 특정 CAN 메시지를 전송한 ECU를 식별하기 위해 확인하고자 하는 ECU의 Error 상태를 강제로 변경시키고, 이러한 상태를 확인하여 해당 메시지를 전송한 ECU를 식별하는 방법이다. ECU의 Error 상태를 강제로 천이시키기 위해 특정 CAN 메시지가 CAN Bus에 전송 중일 때, 해당 메시지를 대상으로 반복적인 Error를 발생시켜 해당 메시지를 전송한 ECU를 원하는 Error 상태로 천이시킨다. 천이시킨 Error 상태를 확인하기 위해 대표적으로 두 가지 Error 상태를 활용할 수 있다. 첫 번째는 Bus-off 상태이며, 두 번째는 Error-passive 상태이다. Bus-off 상태는 ECU의 메시지 송수신이 중지되기 때문에 해당 메시지를 전송한 ECU에서 전송하는 다른 CAN 메시지도 식별할 수 있다[18]. 두 번째 방법은 해당 메시지를 전송한 ECU를 Error-passive 상태로 천이시키고 suspended time interval을 활용하여 해당 ECU에서 전송하는 다른 CAN ID의 식별을 수행한다[21].

Ⅳ. ECU 식별 기법들의 비교 분석 및 한계점

앞서 설명한 CAN 상에서의 ECU 식별 기법의 비교는 [표 2]와 같다. CAN Bus에는 주기적인 메시지 외에도 이벤트 메시지를 포함한 다양한 비주기(aperiodic) 메시지가 존재한다. ECU 식별 기술은 메시지의 전송 유형에 상관없이 모든 메시지에 대해 식별이 가능해야 한다. 또한 ECU 식별을 위한 머신러닝 모델을 학습하기 위해서는 대상 자동차의 내부 네트워크에 대한 사전 정보가 요구된다. 자동차의 기존 내부 정보 없이도 식별이 가능해야 한다. 또한 자동차의 주행 환경은 매우 다양하게 변화하기 때문에 극한의 환경에서도 식별 정확도가 저하되어서는 안 된다. 다양한 환경에서도 일관되고 신뢰성 있는 식별 결과를 얻을 수 있어야 한다.

Voltage 신호를 이용하여 식별하는 방법은 공격자가 가장 모방하기 어렵다는 것이 특징이며 주기 메시지 외에 비주기 메시지에 대해서도 식별이 가능하다는 장점이 있다. 그러나 Voltage 신호를 수집하기 위해 오실로스코프와 같은 전용 장치가 필요하며 온도

〔표 2〕 ECU 식별 기법들의 비교

ECU identification methodologies		Msg type	Prior information	Environment	Re-training	Disruptiveness
Voltage signal	Statistic features	Periodic and aperiodic	Need	Sensitive	Need	Non-disruptive
	Propagation Delay	Periodic and aperiodic	Need	Sensitive	Need	Disruptive
Clock-skew	Message arrival time	Only periodic	Not need	Not sensitive	Not need	Non-disruptive
	Bit time	Periodic and aperiodic	Not need	Sensitive	Need	Non-disruptive
Error state		Periodic and aperiodic	Not need	Not sensitive	Not need	Disruptive

와 습도, 그리고 자동차의 연식과 같은 외부 환경에 따라 Voltage 신호가 변할 수 있다는 한계점이 존재한다. 따라서 이러한 변화를 학습하기 위해 지속적인 재학습(re-training)이 필요하며 재학습 과정에서 공격자에 의해 Voltage 신호가 오염될 수 있다는 연구 결과가 최근 발표되었다[22]. 또한, Voltage 신호와 ECU 간의 매칭을 학습하기 위하여 사전 맵핑 정보를 알고 있어야 한다[19].

Clock-skew 방법 중 메시지 수신 시간에 기반한 방법은 Voltage 신호와 달리 수집을 위한 추가적인 전용 장치가 필요 없으며 온도나 연식과 같은 외부 환경에 강인하다는 장점이 있다. 그러나 수신 시간 기반이므로 비주기 메시지에 대해서는 식별을 수행할 수 없으며[19], 공격자가 성공적으로 특정 ECU의 Clock-skew를 모방하여 공격 메시지 전송이 가능함이 발표되었다[23]. Bit time으로부터 Clock-skew를 추출하는 방법은 Voltage 신호를 이용하는 방법과 동일하게 비주기 메시지에 대한 식별이 가능하고 공격자가 모방하기 어렵다는 장점이 있지만, Voltage signal을 바탕으로 Clock-skew를 추출하기에 외부 환경에 민감하다는 한계점이 있다.

ECU의 Error 상태에 기반하여 ECU를 식별하는 방법은 고유한 하드웨어적인 특징에 구애받지 않고 시스템적인 상태를 검증하는 방식으로 식별을 수행하기 때문에 환경에 영향을 받지 않는다는 장점이 있다. 또한 메시지의 주기성에 상관없이 식별이 가능하다는 장점이 있다. 그러나 식별을 위해 CAN Bus에 실시간으로 Error를 발생시키고 ECU를 특정 Error 상태로 천이시키기 때문에 시스템에 악영향을 미칠 수 있다.

Ⅴ. 결 론

본 고에서는 CAN에서의 악의적인 메시지를 전송한 ECU를 식별하는 ECU 식별 기술에 대해 분석하고 각 기술의 한계점에 대하여 알아보았다. 특히, ECU마다의 고유한 하드웨어적 특징과 수집 방법, 활용 방법에 따라 분류하여 설명하였고, 하드웨어적인 특징을 활용하지 않는 식별 기술에 대해서도 다루었다. 현재까지 발표된 다양한 식별 기술들은 나름의 장점을 보유하고 있지만, 각각 실제 차량에 바로 적용하기에는 여러 한계점도 있었다. Voltage 신호를 활용하는 기술은 공격자가 모방하기 어렵다는 장점이 있지만, 외부 환경에 민감하다는 한계점이 있다. Clock-skew를 활용하는 방법은 비주기 메시지에 대해서는 식별할 수 없다는 대표적인 한계점이 존재하였다. ECU의 Error 상태에 기반하여 ECU를 식별하는 연구는 시스템에 직접적인 영향을 미치기 때문에 추가적인 실제 차량 검증이 필요한 연구이다.

참 고 문 헌

[1] ISO, ISO. “11898-1: 2003-Road vehicles–Controller area network.” International Organization for Standardization, Geneva, Switzerland, 2003.

[2] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham et al., “Experimental security analysis of a modern automobile,” in Security and Privacy (SP), 2010 IEEE Symposium on. IEEE, 2010, pp. 447–462.

[3] Miller, Charlie, and Chris Valasek. “Remote exploitation of an unaltered passenger vehicle.” Black Hat USA 2015, no. S 91, 2015.

[4] Nie, Sen, Ling Liu, and Yuefeng Du. “Free-fall: Hacking tesla from wireless to can bus.” Briefing, Black Hat USA, 25, p.1-16, 2017.

[5] Woo, Samuel, Hyo Jin Jo, and Dong Hoon Lee. "A practical wireless attack on the connected car and security protocol for in-vehicle CAN." Intelligent Transportation Systems, IEEE Transactions on 16.2 (2015): 993-1006.

[6] A.-I. Radu and F. D. Garcia, “Leia: A lightweight authentication protocol for can,” in European Symposium on Research in Computer Security. Springer, 2016, pp. 283–300.

[7] A. Taylor, N. Japkowicz, and S. Leblanc, “Frequency-based anomaly detection for the automotive can bus,” in 2015 World Congress on Industrial Control Systems Security (WCICSS). IEEE, 2015, pp. 45–49.

[8] H. M. Song, H. R. Kim, and H. K. Kim, “Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network,” in 2016 international conference on in-

formation networking (ICOIN). IEEE, 2016, pp. 63 – 68.

[9] M. R. Moore, R. A. Bridges, F. L. Combs, M. S. Starr, and S. J. Prowell, “Modeling inter-signal arrival times for accurate detection of can bus signal injection attacks: a data-driven approach to in-vehicle intrusion detection,” in Proceedings of the 12th Annual Conference on Cyber and Information Security Research. ACM, 2017, pp. 1 – 4.

[10] Stachowski, Stephen, Ron Gaynier, and David J. LeBlanc. An assessment method for automotive intrusion detection system performance. University of Michigan, Ann Arbor, Transportation Research Institute, 2019.

[11] Kneib, Marcel. "A survey on sender identification methodologies for the controller area network." SICHERHEIT 2020 (2020).

[12] Choi, Wonsuk, et al. "Identifying ecus using inimitable characteristics of signals in controller area networks." IEEE Transactions on Vehicular Technology 67.6 (2018): 4757-4770.

[13] Kneib, Marcel, and Christopher Huth. "Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018.

[14] Choi, Wonsuk, et al. "Voltageids: Low-level communication characteristics for automotive intrusion detection system." IEEE Transactions on Information Forensics and Security 13.8 (2018): 2114-2129.

[15] Biham, Eli, Sara Bitan, and Eli Gavril. "TCAN: authentication without cryptography on a can bus based on nodes location on the bus." Proceedings of the 2018 Workshop on Embedded Security in Cars (ESCAR). Vol. 16. 2018.

[16] Moreno, Carlos, and Sebastian Fischmeister. "Sender Authentication for Automotive In-Vehicle Networks through Dual Analog Measurements to Determine the Location of the Transmitter." ICISSP. 2019.

[17] Cho, Kyong-Tak, and Kang G. Shin. "Fingerprinting electronic control units for vehicle intrusion detection." 25th USENIX Security Symposium (USENIX Security 16). 2016.

[18] Kulandaivel, Sekar, et al. "{CANvas}: Fast and Inexpensive Automotive Network Mapping." 28th USENIX Security Symposium (USENIX Security 19). 2019.

[19] Zhou, Jia, et al. "A Model-Based Method for Enabling Source Mapping and Intrusion Detection on Proprietary Can Bus." IEEE Transactions on Intelligent Transportation Systems (2022).

[20] Zhou, Jia, et al. "Btmonitor: Bit-time-based intrusion detection and attacker identification in controller area network." ACM Transactions on Embedded Computing Systems (TECS) 18.6 (2019): 1-23.

[21] Shin, Jiwoo, et al. "RIDAS: Real-time identification of attack sources on controller area networks."

[22] Bhatia, Rohit, et al. "Evading Voltage-Based Intrusion Detection on Automotive CAN." NDSS. 2021.

[23] Ying, Xuhang, et al. "Shape of the cloak: Formal analysis of clock skew-based intrusion detection system in controller area networks." IEEE Transactions on Information Forensics and Security 14.9 (2019): 2300-2314.

〈저 자 소 개〉

이 세 영 (Seyoung Lee)

종신회원

2014년 2월 : 서울시립대학교 수학과 졸업

2016년 2월 : 고려대학교 정보보호대학원 정보보호학과 석사

2016년 3월~현재 : 고려대학교 정보보호대학원 박사과정

<관심분야> 자동차 보안, CPS보안, IoT 보안

최 원 석 (Wonsuk Choi)
종신회원

2008년 2월 : 서울시립대학교 수학과 졸업
2013년 2월 : 고려대학교 정보보호대학원 정보보호학과 석사
2018년 8월 : 고려대학교 정보보호대학원 정보보호학과 박사
2018년 9월~2020년 2월 : 고려대학교 정보보호연구원 연구교수
2020년 3월~2023년 2월 : 한성대학교 IT융합공학부 조교수
2023년 3월~현재 : 고려대학교 정보보호대학원 조교수
<관심분야> 자동차 보안, IoT 보안, 암호학

이 동 훈 (Dong Hoon Lee)
종신회원

1983년 8월 : 고려대학교 경제학과 졸업
1987년 12월 : Oklahoma University 전산학과 석사
1992년 5월 : Oklahoma University 전산학과 박사
1993년 3월~1997년 2월 : 고려대학교 전산학과 조교수
1997년 3월~2001년 2월 : 고려대학교 전산학과 부교수
2001년 3월~현재 : 고려대학교 정보보호대학원 교수
<관심분야> 암호 프로토콜, 암호이론, USN 이론, 키 교환, 익명성 연구, 자동차 보안, IoT 보안