Control Area Network 보안 기술 동향

김상효
성균관대학교
전자전기공학부

김영식
조선대학교 정보통신공학과

Ⅰ. 서 론

오늘날 차량 운행 안전 및 차량 제어의 효율화, 자동화 서비스를 제공하기 위해서 자동차에 많은 전자 장치 및 통신 장치가 사용되고 있다. 특히 최고급 차량뿐만 아니라, 전기 구동 모터와 내연엔진을 함께 사용하는 하이브리드차, 그리고 향후 전기자동차에 이르기까지 다양한 통신 장치를 갖춘 차량들이 점차로 늘어나고 있다. 차량 운행의 안전성 및 효율성을 높이고 사용자에게 편리한 실시간 주행 정보 제공을 위해서 차량 간 (Vehicle-to-Vehicle, V2V), 차량과 기반시설 간 (Vehicle-to-Infrastructure, V2I) 통신을 위한 VANET (Vehicular Ad Hoc Network) 기술이 연구되었고 더 나아가 최근에는 사물인터넷 시대에 맞춰 커넥티드 카(connected car)에 대한 관심이 높아지고 있다.

CAN은 차량 내부의 제어 장치의 증가로 인해 내부의 배선 간소화 및 효율화를 위해 도입되었다. 현재에는 차량 뿐만 아니라 선박, 철도, 항공우주 분야 및 의료기기 제어를 위해 활용되고 있다.

특히 오늘날 자동차 전장 시스템은 네트워크로 연결된 ECU (electronic control units)에서 동작하는 소프트웨어가 핵심 부품으로 사용되고 있다. 이 때 ECU는 직렬 버스 형태의 내부 네트워크를 통해서 센서, 구동기(actuator), 및 ECU 간 데이터 전송을 수행한다. 이 때 사용하는 차량 내 네트워크를 CAN(control area network)이라 부른다.

CAN은 1985년 독일의 Bosch 사에서 차량 부품간 통신을 위한 네트워크로 처음 개발하였다. 과거에 차량에서 사용되는 전자 컨트롤러의 수가 제한적일 때는 컨트롤러간 메시(messy) 형태의 직접 와이어 연결을 통해서 구현하는 것이 가능했지만, 차량의 발전으로 사용되는 컨트

롤러, 센서, 구동기의 수가 크게 증가함에 따라, 관련된 배선의 무게 및 필요한 공간이 증가하고 그에 따른 비용이 크게 증가하였기 때문에 간단하지만 효율적인 네트워크 구조가 개발된 것이다. CAN은 매우 단순하면서도 효율적인 구조를 갖고 있었기 때문에 자동차 업계에서는 그 이후 CAN을 실차 생산에 직접 도입하였으며 1993년에는 CAN 표준인 ISO 11898를 제정하였다[1].

또한 다년간 차량에서 다년간 안정적으로 동작하는 것이 입증된 후에는, 차량뿐만 아니라 선박 내부 통신망이나, 트램, 지하철, 경전철 등 철도, 그리고 승강기와 에스컬레이터에서도 CAN 프로토콜이 응용되며, 공장 설비의 센서와 구동기를 연결하는 통신 네트워크로도 널리 사용되고 있다. 그리고 항공 센서, 항법장치, 항공기내 데이터 분석에서 엔진 컨트롤 시스템 등 우주항공 분야 및 의료기기 임베디드 네트워크로도 CAN을 사용한다. 병원에서 조명, 테이블, X선 기계, 환자 침상 조절 등 수술실 관리를 CAN 기반으로 운용하고 있다.

그러나 CAN이 개발되거나 표준으로 제정될 당시에는 차량 네트워크에 대해 오늘날과 같은 해킹 공격이나 보안 위협이 있을 것을 상정하기가 어려웠기 때문에, 현재 사용되는 CAN에는 암호화나 인증과 같은 암호학적 보안 메커니즘이 포함되어 있지 않았다. 실제로 CAN이 사용된 지 수십 년 동안 해커들도 차량 네트워크에는 거의 주목을 하지 못하였고, 예외적으로 차량 성능을 임의로 개조하고 변경하기 위한 애프터마켓 튜닝 커뮤니티에서만 비교적 관심을 보여왔다.

현재 CAN 프로토콜에는 암호화나 인증과 같은 보안 메커니즘이 포함되어 있지 않다. 그러나 2010년이후 CAN 프로토콜이 해킹에 취약하고 차량 안전에 위협이 된다는 보고가 지속되고 있다.

그러나 2010년에 미국 UCSD와 워싱턴 대학의 Koscher 연구팀은 차량용 네트워크에 대한 해킹 취약성에 대한 종합적인 연구를 수행하였고, 그 결과 차량에 사용되는 네트워크에 대해서 정보통신기기에서와 마찬가지의 해킹 공격이 가능할 뿐만 아니라 차량의 안전운행에 심각한 위협이 될 수 있음을 직접 시연하였다[2-3]. 이를 계기로 차량 내부 네트워크에 대한 취약성에 대한 관심이 고조되었으며, 이에 대한 대응 방법에 대한 연구도 많은 관심을 받아 왔다.

본 논문에서는 CAN 프로토콜에 대한 보안 메커니즘 연구에 대한 동향 및 향후 발전 방향에 대해서 전망해 보고자 한다. 먼저 CAN의 구조에 대한 간략한 소개와 함께, CAN 특징에 따른 보안 취약점 및 보안 요구 사항에 대해서 살펴보고, 마지막으로 CAN 보안을 위해 제안된 대표적인 메커니즘을 살펴볼 것이다.

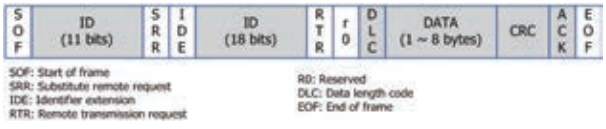
Ⅱ. CAN 프로토콜 개요

이 절에서는 보안 취약점 및 메커니즘을 설명하기 위해서 CAN 프로토콜에 대한 간략한 설명을 제공할 것이다.

CAN에서 이루어지는 통신은 브로드캐스팅 방식이다. 즉, 한 장치에서 전송한 메시지는 버스에 연결된 모든 장치가 수신할 수 있다. 네트워크상의 모든 장치는 전송되는 모든 메시지를 확인한다. 자신에게 전송되는 메시지인지 아니면 필터링해서 무시해야할지를 결정한다.

모든 메시지에는 수신 ID에 따라 우선순위가 정해져 있다. 동시에 메시지를 전송하는 장치가 있다면, 우선순위가 높은 메시지가 먼저 전송되고 낮은 우선순위 메시지는 나중으로 연결된다. CAN ID는 전송되는 메시지 종류를 나타낸다. 따라서 컨트롤러는 브로드캐스트 되는 모든 메시지를 모니터링하면서, 자신과 관련된 메시지인지를 판단하여 디코딩하게 된다.

CAN 물리 계층으로는 크게 고속(high speed) CAN 과 저속(low speed)/내고장(fault tolerant) CAN이 있다. 이 중에서 고속 CAN은 CAN C또는 ISO 11898-2로도 불리며 가장 널리 사용되는 계층으로 두 개의 와이어로 버스가 형성되며 최대 1Mbps의 전송 속도를 지원한다. 저속/내고장 CAN은 CAN B또는 ISO



〈그림 1〉 CAN 프레임 구조

11898-3으로도 불리며 두개의 와이어로 버스가 형성되고 최고 125kbps 속도가 지원된다. 이외에도 최고 33.3 kbps 또는 88.3 kbps 속도를 지원하는 단일 와이어 CAN도 있다.

CAN의 기본 데이터 단위는 프레임이다. 프레임의 주요 필드로는 〈그림 1〉과 같이 수신 ID, 데이터 필드 길이, 데이터 바이트, CRC (cyclic redundancy check), ACK (acknowledgement) 등으로 구성되어 있다. 수신 ID는 표준 11비트가 기본으로 사용되며, 옵션에 따라 18비트가 추가 되어 확장 29비트로 사용할 수도 있다. 또한 수신 ID 크기에 따라 프레임의 우선순위가 결정된다. 사용자 데이터는 바이트 단위로 1에서 8바이트까지 가능하고, 따라서 한 프레임에서 전송 가능한 최대 데이터 크기는 64비트이다. 그리고 16비트 CRC와 ACK 신호가 있는데, 16비트 CRC로 수신된 메시지에 오류가 없는지 검사하게 되고, 메시지를 정확하게 수신한 컨트롤러는 ACK를 통해 정확히 수신되었음을 알려준다. 처음에 메시지를 송신한 컨트롤러는 향후에 해당 ACK를 수신하지 못하면, 다른 설정이 없는 한 해당 메시지를 재전송한다. 실제 사용되는 CAN ID와 데이터 구조는 차량 진단을 위해 예약된 진단용 ID를 제외하고 많은 경우 표준에 지정되지 않았고, 대부분 제조사들이 독자적인 방식으로 정의해 사용하고 있다.

Ⅲ. CAN 프로토콜 보안 위협

차량용 전장 시스템에서 보안은 점점 중요해지고 있다. 공격에 저항하기 위해서 보안 메커니즘이 적용되어야 하지만, CAN은 보안에 대한 고려 없이 설계되어 있기 때문에, 기존의 장치들과의 호환성을 유지하면서 보안 메커니즘을 적용시키는데는 여러 가지 문제가 따른다. 이 절에서는 CAN 프로토콜의 보안 취약성 및 보안 위협에 대해서 살펴보고자 한다.

1. CAN 공격 시나리오

차량 운행 안전 및 차량 내 운전 환경 및 지능 시스템의 안전도를 높이기 위해서 자동차는 많은 정보통신기기를 도입하는 형태로 진화되고 있다. 오늘날에는 자동차도 인터넷 및 다양한 이동 단말에 연결되어 무선으로 진단 및 차량 소프트웨어를 업데이트하고 자동 충돌 방지 시스템 및 음성 제어와 같은 시스템을 동작시킬 수 있다. 이에 따라 차량 시스템에 BlueTooth, 3G/4G, GPS, WiFi, 그리고 무선 센서 통신과 같은 많은 외부 인터페이스가 사용되고 있다.

이런 연결성의 확대는 공격자에게 차량 시스템으로 뚫고 들어갈 새로운 기회를 제공해 준다. 예를 들어 공격자가 블루투스나 이동 통신 연결을 통해 차량 내 전자 제어 장치 중 하나인 차량의 텔레매틱스 장비에 악성 코드를 심을 수 있다.

CAN에서는 안전과 관련된 장치들이 연결된 네트워크에, 안전과 관련성이 적은 인포테인먼트, USB, HMI 등 외부 장비를 연결을 위한 기기들이 게이트웨이를 통해 직간접적으로 연결될 수 있다는 문제가 있다.

보안 관점에서 차량 내 네트워크의 가장 큰 문제점은 안전과 관련된 장치들이 연결된 버스와 안전과 상관없는 장치들이 연결된 버스 사이의 물리적 연결이 존재한다는 점이다. 예를 들어, 속도와 엔진을 제어하는 장치들이 연결된 네트워크와 셀통신, USB, HMI(human machine interface)을 연결하는 네트워크가 게이트웨이를 통해서 연결되어 있다.

하나의 ECU만 보안이 손상되어도 공격자는 가짜 메시지를 차량에 주입함으로써 차량을 제어하는 것이 가능하다. CAN에 연결된 대부분의 컨트롤러들은 외부에 노출되기 때문에, 공격자는 여러 개의 ECU 중 하나를 공격하여 제어 권한을 확보할 수 있다. 그 이후에 CAN 버스에 직접 접근이 가능하여 위조된 메시지를 전송하거나 다른 컨트롤러가 전송했던 정상적인 프레임을 그대로 재전송하는 것이 가능하다. 또는 공격을 위해 설계된 인가되지 않은 장치를 CAN 버스에 직접 연결할 수도 있다.

이런 상황 하에서 다음과 같은 공격이 가능해 진다. 먼저 공격자는 CAN 버스 상에서 전송되는 모든 프레임의 내용을 읽을 수 있다. 버스에서 전송되는 데이터와 그에 따른 시스템의 변화를 감지하면서 프레임에 전송되는 데

이터의 값과 의미를 분석하고 추정할 수 있다. CAN ID는 전송되는 데이터의 종류를 나타내므로 누가 수신하는지를 나타낼 수 있지만, 송신한 장치에 대한 정보가 없기 때문에, 인가되지 않은 장치에서 마치 인가된 장치에서 프레임이 만들어져 전송된 것처럼 위장하는 스푸핑(spoofing) 공격이 매우 용이하다. 더 나아가 CAN 버스를 통해서 정상적인 노드들의 데이터 혹은 소프트웨어를 업데이트할 수도 있다.

2. CAN 보안 요구사항

CAN에서의 보안 위협에 대항하기 위해서는 다음과 같은 일반적인 방어 수단이 존재한다. 먼저 인가되지 않은 장치에서 데이터를 읽어서 내용을 분석하는 것을 방지하기 위해서는 일반적인 암호화/복호화를 사용할 수 있다. 이 경우 ECU 간 혹은 여러 개의 ECU로 구성된 하나의 그룹 내에서 하나의 비밀키가 사전에 공유되어 있어야 하고, 안전한 키 관리를 위한 키 분배서버 및 시간동기화를 위한 시간 서버 등의 역할을 하는 컨트롤러가 지정되어야 한다.

CAN에 보안 인증을 추가하는 경우 장치간 신원확인, 전송되는 메시지의 인증 및 무결성 보장이 중요하다.

인증되지 않은 장치가 데이터를 위조하거나 정상 장치가 전송하는 것으로 위장하는 것을 방지하기 위해서 메시지 인증 코드(message authentication code)가 사용될 수 있다. 메시지 인증 코드 생성을 위해서 해시 함수와 같은 안전한 암호학적 알고리즘이 사용되어야 하고, 관련된 ECU간 혹은 ECU 그룹에 대한 키를 관리할 수 있는 메커니즘이 제공되어야 한다. 키 분배가 안전하게 이루어지면, ECU에서는 전송되는 메시지에 대한 메시지 인증 코드를 비밀키를 이용해서 생성하여 메시지에 연접하여 전송하게 된다.

경우에 따라 송신자와 수신자가 상대방의 신원을 확인할 수 있도록 하는 인증 기능이 필요하다. 이를 위해 사전에 공유된 비밀 키를 활용하여 인증이 이루어지게 된다. 이 때 물리적 복제방지함수(physically unclonable function)를 사용하면 ECU 상에서의 키 생성 및 인증을 강화하는 것이 가능하다[4].

차량 내부 통신을 위해서는 데이터 무결성과 인증이 중요한 보안 요구 사항이 된다. 그러나 내부의 제어 신호나 중요한 상태 정보를 보호하고 임의적 위변조를 막기 위해서는 기밀성 역시 고려되어야 한다.

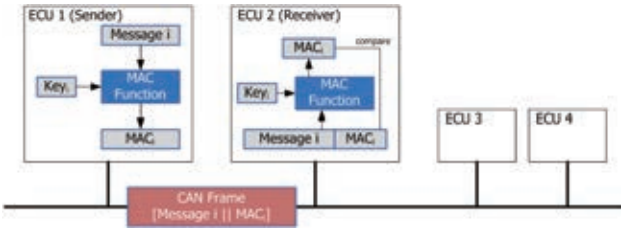
그러나 일반적인 암호학적 솔루션들은 상대적으로 다른 연산들에 비해 높은 계산량이 필요하지만, 차량용 임베디드 시스템들은 많은 경우 8비트나 16비트 코어에 제한적인 메모리만 갖추고 있기 때문에, 이런 계산을 제한 시간 내에 끝낼 수가 없다. 특히 암호학적 연산이 많아지면 본래 해야 할 센서와 구동기 제어 및 통신 관련 기능을 위한 작업에 방해가 되어 실시간 처리가 어려워질 수 있다. 따라서 일반적으로 암호학적 연산을 별도로 수행하는 HSM(hardware security module)과 같은 보안 장치를 추가적으로 도입해야 한다[5].

CAN에서의 메시지 인증을 위한 메커니즘들은 추가로 다음과 같은 요구사항들을 만족시켜야 한다.

- 재전송공격에 대한 저항성 : 이전에 전송된 인증된 메시지를 재전송하는 경우, 컨트롤러에서는 이런 메시지를 검출할 수 있어야 한다.
- 그룹 키 지원 : 키 저장 공간 크기를 줄이기 위해 여러 ECU로 구성된 하나의 그룹에서 하나의 키를 통한 인증이 지원되어야 한다.
- 호환성 : 새로운 인증 메커니즘은 인증 기능이 없는 기존의 컨트롤러에서도 문제를 일으키지 않아야 한다. 이를 위해 CAN 프로토콜의 구조는 변경되지 말아야 한다. 그러나 다음에서 논의하는 것처럼 이런 요구사항을 만족시키는 것은 단순한 문제가 아니다.

3. 왜 CAN 보안이 단순하지 않은가?

일반적으로 CAN에서는 공격자에 의해서 임의적인 데이터 주입에 의해서 동작이 교란될 수 있기 때문에 데이터 무결성 및 인증이 가장 중요하다. 이를 위해서 각 컨트롤러 쌍, 혹은 특정 그룹 내의 컨트롤러들은 하나의 마스터 비밀키를 공유하고 있는 것으로 가정한다. 게이트웨이나 연산능력이 상대적으로 높은 컨트롤러는 키 분배 서버



〈그림 2〉 CAN에서의 메시지 인증 과정

역할을 담당하여 엔진 시동 단계에서 각 장치에 새로운 세션키를 분배한다. 또는 컨트롤러의 요청에 의해서 새로운 세션키를 분배할 수도 있다.

송신자는 인증을 위해서 전송되는 메시지에 대한 메시지 인증 코드(MAC)를 공유된 비밀 세션키를 사용해서 계산하고, 이 값은 메시지에 붙어서 함께 브로드캐스팅 된다. 그러면 모든 수신자들은 공유된 키를 사용하여 수신된 메시지로부터 MAC을 계산한 후, 수신된 MAC과 자신이 계산한 MAC이 같은지를 비교함으로써 인증을 수행할 수가 있다. 이런 동작 과정이 〈그림 2〉에 도시되어 있다.

그러나 CAN 통신에서는 이런 일반적인 인증 방식을 직접 사용하는데 큰 문제가 뒤따른다. 첫 번째로 통신을 위한 오버헤드가 너무 크다는데 있다. 사전에 비밀키를 나누어 가진 경우에 모든 ECU들은 통신에 필요한 각 쌍의 컨트롤러, 혹은 여러 컨트롤러에 대해서 비밀키를 소지하고 관리해야 한다. 또한 안전한 비밀키 사용을 위해서 비밀키는 주기적으로 업데이트 되어야 한다. 이 때 비밀세션키 교환을 위한 프로토콜은 많은 프레임들을 각 컨트롤러 사이에서 전송하도록 해야 한다. 비밀키 교환을 위한 메시지는 보안 프로토콜에 참여하는 송신노드 수가 n개이고 수신 노드의 수가 m개일 때 nm개의 송수신 쌍에 대해서 교환이 이루어 져야 한다. 따라서 모든 ECU가 비밀키 보안에 참여하는 것이 아니라 운행 및 안전에 핵심적인 역할을 하는 컨트롤러들만 키 교환에 참여하도록 만들어야 한다.

또한 인증을 위한 메시지 인증 코드를 생성하기 위해서 SHA-2, SHA-3와 같은 암호학적 해시 함수를 사용할 수 있다. 그러나 대부분의 ECU 들은 제한 시간 내에 암호학적 해시 연산을 수행하기에 충분한 연산능력을 갖고 있지 못하다. 이를 위해 HSM을 추가하여 암호학적 연산을 보조할 수 있다.

가장 큰 문제점은 CAN에서 지원 가능한 데이터의 크기가 최대 64비트이기 때문에 적어도 32비트 이상 되는 MAC을 추가하는 것은 사실상 CAN의 처리율을 반 이하로 줄이는 결과를 낳게 된다.

이와 같은 이유들로 인해서 CAN을 위한 인증 메커니즘들은 제한적인 환경 내에서의 인증을 제공할 수 있는 것들이어야 한다. 따라서 기존의 인증 프로토콜들을 CAN 버스에 그대로 적용하기에는 몇 가지 어려운 점이 있다. 특히 기존의 다른 프로토콜과의 호환성을 고려한 설계로 인해서 다음과 같은 제한 조건을 만족해야만 한다.

- 엄격한 실시간 처리 : CAN이 적용되는 많은 시스템들은 많은 작업들이 엄격하게 실시간 처리를 준시해야만 한다. 따라서 메시지 인증을 위한 프로세스가 기존의 실시간 작업에 영향을 주어서는 안된다.
- 메시지 길이 : CAN 버스의 메시지는 최대 8바이트가 한계이다. 추가적인 인증 데이터는 다른 프레임을 통해서 재전송될 수 있다.
- 메시지 ID : CAN에서는 11비트 또는 29비트의 ID가 사용될 수 있는데, 이미 많은 ID들이 정의되어 사용 중에 있다. 따라서 인증을 위해 추가되는 ID들은 제한적으로 주의 깊게 다루어져야 한다.
- 단방향 통신 : CAN 버스로 전송되는 프레임에는 송신자에 대한 정보가 들어 있지 않으며, 모든 정보는 브로드캐스팅된다. 따라서 컨트롤러 사이의 양방향 통신은 고려되지 않고 있다. 예외적으로 오류 플래그를 통해서만 반대 방향 전송이 가능한데 이 경우에도 어느 컨트롤러에서 오류가 발생했는지를 알아낼 수가 없다.

CAN 보안은 실시간 처리를 고려한 통신 오버헤드 최소화, 하드웨어 보안 모듈(HSM) 기반 암호알고리즘 구동, 그리고 64비트 제한 내에서의 인증 데이터 처리 등이 고려되어야 한다.

메시지 인증을 위해 메시지 인증 코드(MAC)를 계산하여 함께 전송해야 하지만, 엄격한 실시간 제한 조건 때문에, 사용하는 MAC 알고리즘은 빠르게 동작해야 한다. 인

증 데이터의 일부가 도착하자마자 빠르게 검증 프로세스가 시작되어야 한다.

재생공격에 대한 저항하기 위해서는 MAC을 계산할 때 카운터 값이 포함되어야 하며, 동일한 카운터값이 다시 사용되면 안 된다. 하지만 카운터 크기가 제한되어 있고 최대 한계치에 도달하게 되면 문제가 발생하게 된다. 그러므로 인증에 사용되는 비밀키는 시동이 걸리는 단계에서 새로 업데이트되는 세션키 형태로 사용하여야 한다.

메시지 ID와 단방향 통신에 대한 제한 조건은 그들이 수신한 메시지에 대한 응답을 보낼 수 없다는 것이다. 새로운 ID들을 사용해야 한다. 예를 들어 특정 ID는 특정 컨트롤러에 대한 응답으로 사용할 수 있다. n개의 컨트롤러가 m개의 다른 메시지 ID에 대해서 사용될 수 있다면, nm개의 ID가 새로 사용되어야 한다. 실제로는 11비트의 ID로는 달성하기 어렵다.

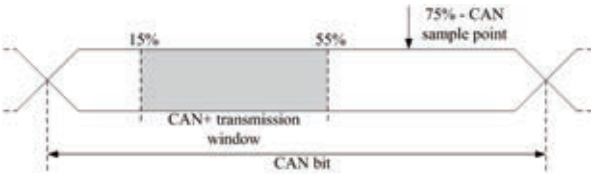
호환성 문제는 가장 심각하다. 존재하는 CAN 메시지에 인증 데이터를 이어 붙여 전송하는 것은 일반적으로 메시지 길이를 위반하기 때문에 불가능하다. 인증 데이터를 여러 메시지에 걸쳐서 전송하는 것은 시스템의 실시간 처리 능력에 영향을 주게 된다.

Ⅳ. CAN 프로토콜 보안

CAN 프로토콜 보안 문제가 대두되면서 CAN 프로토콜을 위한 보안 메커니즘들이 제안되었다. 본 절에서는 CAN을 위한 대표적인 보안 메커니즘들에 대해서 살펴본다.

1. CAN 인증 메커니즘

인증 데이터를 추가하는 문제를 해결하기 위해서 CANauth 프로토콜이 제안되었다[6]. CANauth에서는 CAN+ 프로토콜에서 지원되는 여분의 비트 공간을 활용해서 인증 정보를 전송하게 된다. CAN+에서는 추가적인 비트가 CAN 버스 인터페이스의 샘플링 지점 사이에 추가될 수 있다[7]. 이를 〈그림 3〉과 같이 나타낼 수 있다.



〈그림 3〉 CAN+ 프로토콜에서 데이터를 주입하는 구간

이런 방식으로 전송될 수 있는 비트는 CAN+ 인터페이스의 최대 동작 속도에 의해서 제한된다. Zimermann 등은 1MHz CAN 네트워크에서 300MHz로 동작하는 FPGA에서 15개의 CAN+ 바이트를 각 CAN 바이트 하나당 전송하였다[7]. 더 낮은 CAN 버스 속도에서 이 숫자는 더 커질 수 있다. 이 관계는 다음과 같이 나타낼 수 있다.

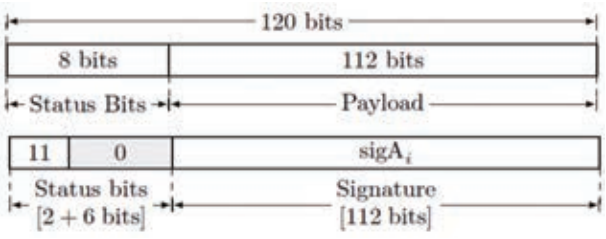
$$\frac{\text{CAN+ data bits}}{\text{CAN data bit}} = \frac{1\text{MHz} \times 16}{f_{bus}} - 1$$

위 식에서 1을 빼 준 것은 모든 CAN+ 전송이 시작될 때 시작 비트를 넣어 주어야하기 때문이다. 100kHz CAN 네트워크에서, 159 CAN+ 비트를 각 CAN bit마다 추가할 수 있다.

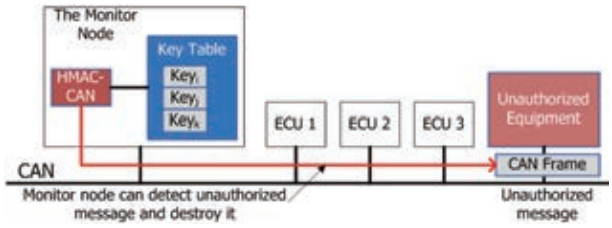
길이나 CAN 버스 네트워크의 속도에 상관없이 메시지를 인증할 수 있어야하기 때문에 인증 데이터는 15바이트로 제한된다. 따라서 매우 긴 키를 사용하는 공개키 기반의 인증은 사용할 수 없다.

CANauth 방식은 CAN+ 프로토콜에 의해서 제공되는 여분의 데이터 공간에 112비트의 인증 비트를 전송하도록 하고 있다. 1MHz의 속도를 갖는 CAN 프로토콜에서는 15바이트의 여분의 비트 전송이 가능하고, 이 중에서 8비트는 상태 비트를 전송하고 나머지 112비트에 인증 값을 전송한다. 실제 112비트 중에서 32비트는 카운터 값이고 나머지 80비트가 MAC값이다. 그러나 이는 CAN+ 프로토콜을 지원할 수 있는 하드웨어가 있어야만 한다.

좀 더 현실적인 CAN에서의 인증 방법이 Hartkopp 등



〈그림 4〉 CANauth 방식에서의 서명 비트



〈그림 5〉 HMAC-CAN 방식 동작 개요

에 의해서 제안되었다[8]. Hartkopp는 64비트의 데이터 크기에 맞추기 위해서 MAC의 길이를 최대한 줄이는 방법을 제안하였다. 엔진이 시동될 때마다 매번 세션키를 새로 업데이트한다고 가정하면, 고속 CAN의 동작 환경하에서는 32비트의 MAC 크기가 안전한 최소 크기로 평가하였다. 또한 양방향 인증이 가능하기 위해서 64비트의 메시지 중 일부는 인증을 위한 ID를 나타내는 용도로 사용하였다.

스푸핑 공격에 대응하기 위해서 정상적인 노드에 허가되지 않은 데이터가 전송되지 않도록 하는 대신, 공격자에 의한 허가되지 않은 메시지를 오류 메시지를 전송하여 덮어씌우는 방법이 제안되었다[9]. 그러나 이 방법에서는 차량 내부의 네트워크에서 모든 정상적인 노드는 특별한 CAN 제어기를 갖고 있어야 한다.

이 문제를 완화하기 위해서 모든 장치들이 특별한 CAN 제어기를 가져야 하는 대신 중앙의 컨트롤러가 MAC을 갖고 있는 메시지를 인증하기 위해 사용될 수 있다[10]. 이때 모니터링을 하는 컨트롤러는 CAN 버스에 속해야 한다. 그리고 모니터링을 하는 커트롤러는 CAN 버스에 속해 있는 컨트롤러들을 인증할 수 있어야 한다. 또한 모니터링을 하는 컨트롤러는 MAC을 가진 모든 메시지를 인증할 수 있어야 한다.

이렇게 되면 특별한 하드웨어는 모니터링을 수행하는 컨트롤러에만 있으면 된다. 이 경우 프로토콜은 하나의 모니터링 컨트롤러와 다른 ECU 사이를 인증하게 된다. 모니터링 컨트롤러는 HMAC-CAN에 다른 ECU들은 변하지 않는다. HMAC-CAN은 에러 프레임을 덮어씌움으로써 인가되지 않은 메시지를 파괴한다.

2. 잠재적인 CAN 보안 이슈

차량이 발전하면서 내부에서 사용하는 소프트웨어의 크기와 중요성이 증가하였다. 소프트웨어 복잡도가 증가하는데 많은 경우 안전 관련 기능 보다는 인포테인먼트 지원 기능이 많이 있다. 또한 차량 진단을 위해 마련된 OBD-II 포트를 통해서 외부에서 네트워크에 직접 연결이 가능하다. 또는 인포테인먼트 시스템을 통해서 차량에 간접적으로 접속할 수 있다. 실제로 2011년에 UCSD와 워싱턴 대학의 Kocsher 연구팀은 MP3 파일 헤더에 주입된 임의의 CAN 프레임을 버퍼오버플로우 방식으로 CAN 네트워크에 주입이 가능하다는 사실을 시연하기도 하였다[3].

차량 내 정보보안이 중요한 이유는 차량은 안전과 직결되는 장치이다. 차량에 대한 해킹이 가능하다는 것은 일반 PC를 해킹 가능하다는 것과는 상황이 다르다. 차량은 소프트웨어를 정기적으로 업데이트하는 것이 일반적이지도 않다. 정비소에 들리기 전에는 쉽게 업데이트를 할 수 없다. 많은 경우에 공격자들이 쉽게 물리적인 접근이 가능하다.

사물인터넷 및 커넥티드카에서는 보안 시스템이 차량 안전에 직접적인 영향을 미칠 수 있으며, 이를 위해서는 오늘날 차량의 내부 네트워크로 널리 활용되는 CAN 보안 기술 확보가 선행되어야 한다.

더 많은 인포테인먼트 장치들이 차량에 기본적으로 탑재되고 있다. 현대의 차량용 소프트웨어는 1억라인 정도의 코드를 담고 있다. 일반적으로 천라인당 1개의 보안 문제가 발생하는데, 대략 10만개 이상의 보안 문제 가능성이 잠재되어 있다. 특히 오랜 세월 동안 차량용 소프트웨어는 보안 위협의 사각지대에서 독자적으로 발전되어 왔기 때문에 IT용 장치에 비해서 문제가 많은 레가시 코드가 많이 있다.

여기에는 직접 차량용 네트워크에 접속하는 방식과 차량의 인포테인먼트 시스템을 통해 접속하는 방식으로 나눌 수 있다.

Ⅴ. 향후 연구 및 결론

지금까지 오늘날 차량을 비롯한 산업용 제어 시스템에

서 다방면으로 활용되고 있는 CAN 프로토콜에 대한 보안 문제를 살펴보았다. 사물인터넷의 등장으로 향후 연결성이 크게 증가하여 사용자의 편의성 및 안전성을 증가시킨 커넥티드카에 대한 관심이 고조되고 있다. 그러나 이를 위해서는 충분한 보안이 뒷받침되어야만 본래의 의도대로 구현되는 것이 가능하고, 공격자에 의해서 악용되어 오히려 안전이 위협받는 문제를 사전에 차단할 수가 있다. 현재에는 기존 장치들과의 호환성 문제로 인해서 초기의 CAN 프로토콜을 중심으로 한 보안 메커니즘 설계가 많은 관심을 받아 왔으며, 중요한 알고리즘들에 대해서 본 논문에서 간략하게 리뷰하였다. 이미 새롭게 출시되는 차량들에서는 기존의 CAN의 용량으로는 한계에 도달하고 있기 때문에, 그 이후의 CAN Version 2.0 프로토콜 및 차량용 이더넷 기술에 대한 보안 기술이 뒷받침되어야 할 것이다.

참고문헌

[1] "Road vehicles — Controller area network (CAN)," ISO 11898, 2003.

[2] K. Koscher, et al., "Experimental security analysis of a modern automible," in Proc. IEEE Symp. Security and Privacy, Oakland, CA, May 16–19, 2010.

[3] S. Checkoway, et al., "Comprehensive Experimental Analyses of Automotive Attack Surfaces," USENIX Security, August 10–12, 2011.

[4] G. Sigl, "Physical Unclonable Functions: Chances and Risks of a new Security Primitive," in Proc. ESCAR 2013.

[5] B. Weyl, et al., "Securing vehicular on–board IT systems: The EVITA Project," in Proc. 25th Joint VDI/VW Automotive Security Conf., Oct. 2009.

[6] A. Van Herrewege, D. Singelee, I. Verbauwhede, "CANAuth – A Simple, Backward Compatible Broadcast Authentication Protocol for CAN bus," in Proc. ESCAR 2011, 2011.

[7] T. Ziermann, S. Wildermann, and J. Teich, "CAN+: A new backward–compatible Controller Area Network (CAN) protocol with up to 16x higher data rates," in DATE. IEEE, 2009, pp. 1088–1093.

[8] Oliver Hartkopp, Cornel Reuber, "MaCAN – Message Authenticated CAN," in Poc. ESCAR 2012, 2012.

[9] T. Matsumoto, et al., "A Method of Preventing Unauthorized Data Transmission in Controller Area Network," Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th, 2012.

[10] R. Kurachi, et al., "CaCAN – Centralized Authentication System in CAN (Controller Area Network)," in Proc. ESCAR 2014.

김 상 효

- 2004년 3월~2006년 8월 삼성전자, 책임연구원
- 2006년 9월~2007년 8월 University of Southern California, Visiting scholar
- 2007년 9월~현재 성균관대학교, 정보통신공학전공, 부교수

〈관심분야〉
무선통신, 부호이론

김 영 식

- 2001년 2월 서울대학교 전기공학부, 공학사
- 2003년 2월 서울대학교 전기컴퓨터공학부, 공학석사
- 2007년 2월 서울대학교 전기컴퓨터공학부, 공학박사
- 2007년 3월~2010년 8월 삼성전자 시스템 LSI 사업부, 책임연구원
- 2010년 9월~현재 조선대학교 정보통신공학과, 조교수

〈관심분야〉
사물인터넷보안, 제어시스템 보안, 포스트양자암호