ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 20-02-42
294

j.inst.Korean.electr.electron.eng.Vol.24,No.2,665~668,june 2020
http://dx.doi.org/10.7471/ikeee.2020.24.2.665

CAN 버스에서 노드 ID 자동 설정을 통한 물리 계층 보안 기법

Physical Layer Security Methodwith CAN Bus Node ID Auto-Setting

강 태 욱*, 이 종 배*, 이 성 수*★

Tae-Wook Kang*, Jong-Bae Lee*, and Seongsoo Lee**★

Abstract

When a node in automotive CAN bus is hacked, such node should be blocked to prevent severe danger in the car. In order to do that, such node should be uniquely identified. However, there is no way to identify individual nodes in a CAN bus. In this paper, a physical layer security method is proposed where individual nodes are identified by assigning unique ID to the nodes during booting process. The proposed method was implemented in a CAN controller using Verilog HDL, and it is verified that the node ID auto-setting and internal attack defense are successfully performed.

요 약

자동차 내부의 CAN 버스에서 노드 하나가 해킹을 당한 경우, 차량에 위해를 가하지 못하게 해당 노드를 차단하려면 각 노드를 고유하게 특정하여야 하지만 CAN 버스에는 이러한 기능이 존재하지 않는다. 본 논문에서는 CAN 버스가 부팅될 때 개별 노드에 고유 ID를 자동으로 부여하는 물리 계층 보안 기법을 제안한다. 제안한 기법을 Verilog HDL을 이용하여 CAN 컨트롤러에 구현하였고, 이를 통해 CAN 버스 노드의 고유 ID가 자동으로 부여되고 악의적인 내부 공격이 차단됨을 확인하였다.

Key words : Controller Area Network, Auto-ID, Physical Layer Security, Internal Attack, Error Counter, Bus Off

Ⅰ. 서론

대부분의 자동차 전장 시스템은 ECU(Electronic Control Unit)에 의해 제어되며 CAN(Controller Area Network) [1]-[3] 버스를 통해 데이터를 주고받는다. 차량에 탑재되는 ECU에는 프로세서 ID와 통신용 암호키가 고유하게 지정되어 있어서 통신을 수행할 때마다 프로그램에서 확인하며 해킹이 의심되는 경우 통신을 차단하도록 명령을 내린다. 그러나 해킹이 고도화되면 해당 ECU의 모든 소프트웨어 동작을 변조할 수 있으므로 CAN 버스 상에서 전송되는 정상적 통신 데이터를 모니터링하여 프로세서 ID 및 통신용 암호키를 유추하고 위조할 수 있으며 통신 차단 명령도 무시할 수 있다. 따

* School of Electronic Engineering, Soongsil University(Graduate Student, Post Doc Researcher, Professor)
★ Corresponding author
E-mail : sslee@ssu.ac.kr, Tel : +82-2-820-0692
※ Acknowledgment
This work was supported by the IT R&D program of Ministry of Trade Industry and Energy (MOTIE)/Korea Evaluation Institute of Industrial Technology (KEIT) (20003771)
Manuscript received May. 31, 2020; revised Jun. 15, 2020; accepted Jun. 19, 2020.

라서 해킹에 의한 통신 방해 및 악의적 데이터 전송을 원천적으로 방지하기 위해서는 해킹된 노드의 통신을 CAN 버스 상에서 하드웨어적으로 차단하여야 한다.

이미 해킹되어 CAN 통신을 방해하고 악의적 데이터를 전송하는 노드를 차단하기 위해서는 가정 먼저 CAN 버스 상에서 개별 노드를 고유하게 특정하여야 하지만 CAN 버스에는 이더넷의 MAC (Media Access Control) 주소와 같이 개별 노드를 식별할 수 있는 기능이 존재하지 않는다[4]. CAN 콘트롤러 생산 시 하드웨어적으로 고유 ID를 부여하는 방법도 있으나, 이 경우 모든 칩마다 고유 ID를 부여하기 위해서는 고유 ID의 비트 수가 크게 늘어나서 CAN 버스의 전송 효율을 크게 떨어뜨린다. 모든 노드가 CAN 버스 상에 장착된 후에 공장에서 일괄적으로 고유 ID를 부여하면 동일 CAN 버스 내의 노드 수가 얼마 안되기 때문에 고유 ID의 비트 수는 크게 줄어들지만 부품 교체 때마다 다시 공장에서 ID 부여 작업을 수행해야 하기 때문에 매우 번거롭다.

본 논문에서는 CAN 버스에서 부팅 때마다 개별 노드에 고유 ID를 자동으로 부여하여 각 노드를 구별하는 물리 계층 보안 기법을 제안한다. 제안하는 기법은 부팅 때마다 고유 ID를 할당하기 때문에 부품 교체와 상관없이 쉽게 적용이 가능하면서도 고유 ID의 비트 수를 크게 줄일 수 있다. 제안한 기법을 CAN 컨트롤러[5]에 Verilog HDL을 이용하여 구현하였으며 시뮬레이션을 통하여 유효성을 검증하였다.

II. CAN 버스에서 내부 공격에 대응하는 기법

본 논문에서는 그림 1과 같이 CAN 버스 공격에 대처한다. 이 기법 자체는 이미 [4]에서 제안된 것으로, 그 내용을 요약하면 다음과 같다. [4]에서는 IDS(Intrusion Detection System), NES(Node Expulsion System)라는 하드웨어를 사용한다. IDS는 기존 연구에서도 많이 제안된 바 있으며 데이터 프레임의 내용을 분석하여 현재 송신 중인 노드가 해킹당한 노드인지 판단한다. NES는 IDS가 해킹당한 노드로 지목한 특정 노드를 CAN 버스에서 추방하는 기능을 수행하는 블록으로 일반적인 CAN 버스에는 없는 기능이다.

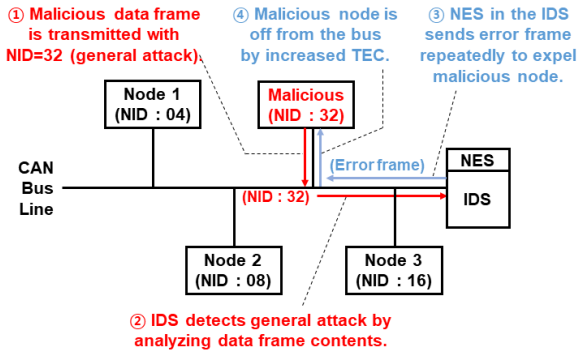

Fig. 1. Coping with CAN bus internal attack.
그림 1. CAN 버스 내부 공격에 대한 대처

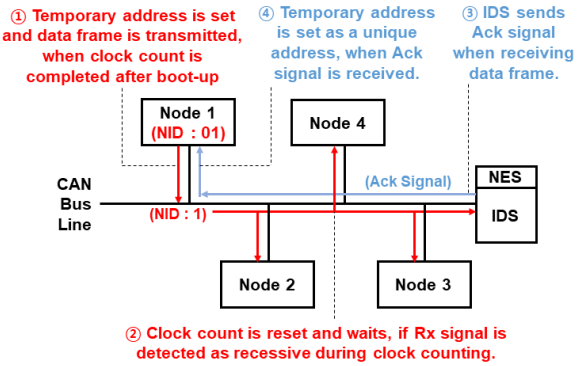

Fig. 2. Node ID auto-setting.
그림 2. 노드 ID 자동 설정

IDS는 CAN 버스를 항상 모니터링하다가 어떤 노드가 악의적인 데이터 프레임을 송신하는 경우 데이터 내용을 분석하여 노드가 해킹당한 것을 감지한다. 이후 해킹당한 노드가 송신할 때마다 데이터 내용에 관계없이 NES가 에러 프레임을 발생시켜 송신을 막는다. 해킹당한 노드는 데이터를 송신할 때마다 송신 에러 카운트가 지속적으로 증가하고, 에러 패시브 상태를 지나 버스 오프 상태가 되어 더 이상 송신이 불가능하게 된다. 본 논문에서는 IDS가 해킹을 감지할 수 있다고만 가정하고 그 방법에 대해서는 다루지 않는다.

[4]의 기법에서는 그림 1에서처럼 CAN 버스 내부의 모든 노드에 고유 ID가 지정되어 있는데, 서론에서 언급했듯이 CAN 콘트롤러마다 생산 시에 고유 ID를 지정하면 고유 ID의 비트 수가 너무 늘어나고, 모든 노드가 차량에 장착된 다음에 고유 ID를 지정하면 부품 교체 시마다 고유 ID를 재지정해야 한다. 본 논문에서는 이러한 문제점을 해결하기 위해 CAN 버스가 부팅할 때 각 노드의 고유 ID를 자동으로 부여하는 방법을 제안한다.

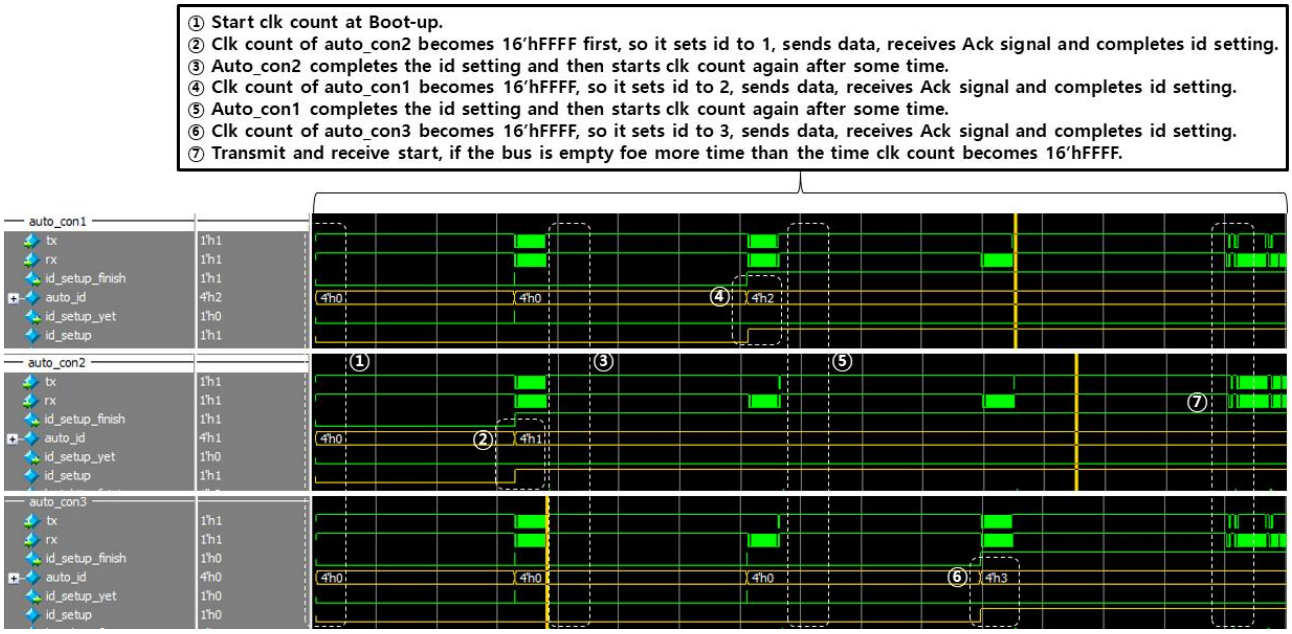

Fig. 3. Simulation of node ID auto-setting when CAN bus is boot-up.
그림 3. CAN 버스가 부팅되면서 노드의 고유 ID가 자동으로 부여되는 과정의 시뮬레이션 결과

III. 노드 ID를 자동으로 설정하는 기법

본 논문에서는 CAN 버스가 부팅될 때 클록 카운트를 기반으로 각 노드의 고유 ID를 설정한다. 그림 2와 같이 모든 노드들은 부팅이 됨과 동시에 클록 카운트를 시작한다. 클록 오실레이터 회로에는 편차가 존재하므로 어떤 노드의 클록 카운트가 먼저 16′hFFFF에 도달하면 그 순간에 해당 노드의 ID를 1로 설정하고 데이터 프레임을 송신한다. 이때 IDS로부터 Ack 신호를 받으면 설정했던 주소를 고유 ID로 설정한다. 반면 클록 카운트가 증가하는 중간에 Rx 신호가 Recessive로 감지되면 클록 카운트를 초기화하고 기다린다. 이때 각 노드들은 클록 카운트의 초기화 횟수를 기억한다. 버스가 휴지 상태가 되면 다시 클록 카운트를 시작하고 클록 카운트가 16′hFFFF가 되는 순간 클록 카운트의 초기화 횟수에 1만큼 더한 값을 계산해 두었다가 IDS로부터 Ack 신호를 받으면 이 계산값을 고유 ID로 설정한다.

제안하는 기법에서는 가장 먼저 클록 카운트가 16′hFFFF에 도달하는 노드가 ID=1이 되고 다른 노드들은 클록 카운트가 초기화되며, 남은 노드 중에 가장 먼저 클록 카운트가 16′hFFFF에 도달하는 노드가 ID=2가 되고 다른 노드들은 역시 클록 카운트가 초기화된다. 이렇게 하여 버스에 연결된 모든 노드들의 주소를 설정한다.

IV. 시뮬레이션 결과

본 논문에서는 기존의 CAN 컨트롤러에 NES를 추가하여 Verilog HDL을 이용하여 구현하였으며 IDEC(IC Design Education Center)에서 제공한 ModelSim으로 시뮬레이션 하였다. 시뮬레이션에서는 두 가지 상황을 확인하였는데 하나는 CAN 버스가 부팅되면서 자동으로 노드의 고유 주소를 부여하는 상황이며 다른 하나는 내부 공격이 이루어지고 이에 대한 방어가 이루어지는 상황이다.

그림 3은 CAN 버스가 부팅되면서 자동으로 노드의 고유 주소를 부여하는 시뮬레이션 결과이다. 처음에 부팅이 시작되고 각 노드들은 클록 카운트를 증가시킨다. 그림 3에서는 가장 먼저 클록 카운트가 16′hFFFF가 된 auto_con2 노드가 고유 ID로 1을 부여받고, auto_con1 노드와 auto_con3 노드는 클록 카운트 중에 Rx 신호가 Recessive로 감지되면서 클록 카운트를 초기화한다. 이때 auto_con1, auto_con3 노드는 클록 카운트의 초기화 횟수(이 시점에서는 1)를 기억한다. 이제 버스가 Idle 상태가 되면 다시 클록 카운트를 시작한다. 이런 식으로 auto_con1 노드와 auto_con3 노드 순으로 고유 ID 설정을 완료한다. 충분한 시간동안 버스가 비어있게 되면 모든 노드들은 ID 부여 작업이 완료되었다고 생각하여 실제 동작을 시작하면서 데이터 프레임의 송수신을 시작한다.

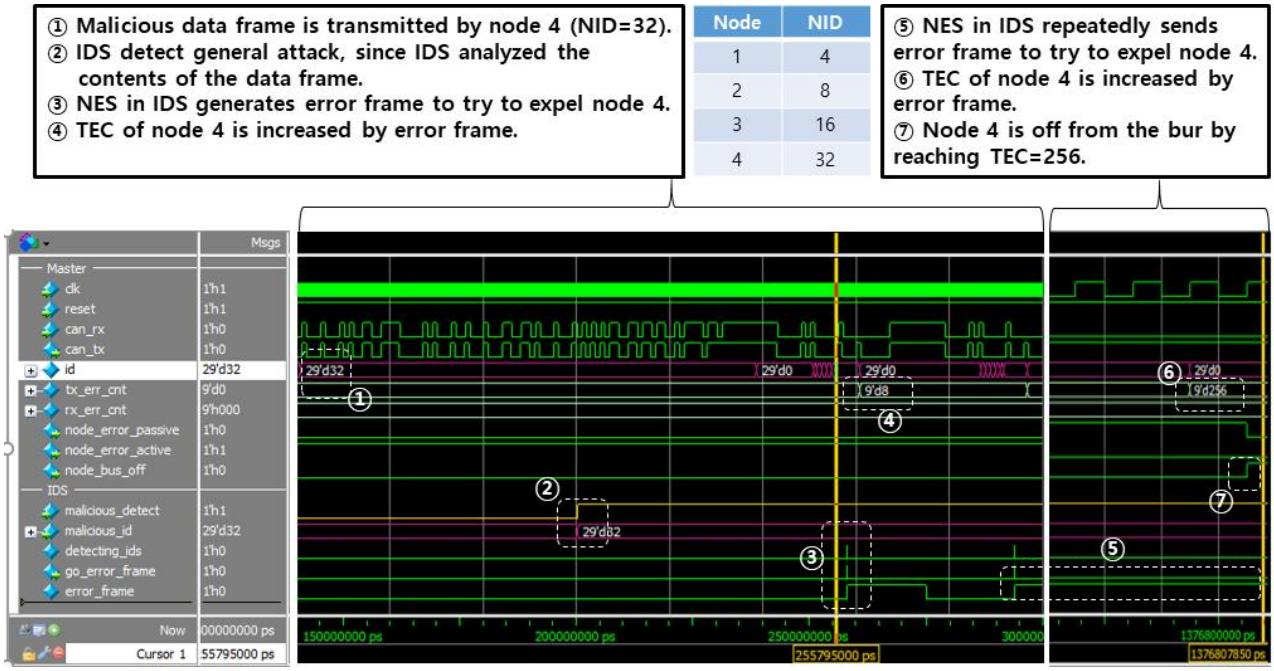

Fig. 4. Simulation results of counterattack against CAN bus internal attack.
그림 4. CAN 버스 내부 공격에 대한 대처 동작의 시뮬레이션 결과

그림 4는 이렇게 각 노드의 고유 ID가 설정된 이후에 내부 공격이 시작되고 이에 대한 대처가 이루어지는 시뮬레이션 결과이다. 이 시뮬에이션에서는 4개 CAN 노드의 ID가 4, 8, 16, 32로 설정되었다고 가정하였다. 그림 4에서는 처음에는 정상 동작을 하다가 일정 시간 후 IDS에서 데이터 프레임의 데이터를 보고 ID가 4인 노드가 해킹당한 것을 감지하였다. 이후에는 ID가 4인 내부 공격 노드가 데이터 프레임을 송신할 때마다 에러 프레임을 발생시켜 해당 노드의 TEC를 증가시킨다. 이 작업은 ID가 4인 내부 공격 노드가 데이터 프레임을 전송할 때마다 이루어지며, 해당 노드의 TEC가 계속 증가하여 결국엔 에러 패시브 상태를 지나 버스 오프 상태가 된다.

V. 결론

CAN 버스 상의 노드에는 주소가 없기 때문에 해킹을 당하여 악의적인 데이터 프레임을 전송하여도 어느 노드가 해킹 당했는지 식별하기 어렵다. 본 논문에서는 기존의 CAN 컨트롤러를 수정하여 CAN 버스가 부팅할 때마다 자동으로 노드의 고유 ID를 정하고 이를 통해 CAN 버스 내부의 공격에서 안전하게 방어하는 기법을 제안하였다. 이를 검증하기 위해 Verilog HDL을 이용하여 CAN 컨트롤러를 설계하였으며 시뮬레이션을 통해 제안한 기법이 성공적으로 노드마다 ID를 자동으로 부여하고 내부 공격에 대한 방어를 수행하는 것을 확인하였다.

References

[1] ISO 11898-1:2015, “Road Vehicles-Controller Area Network (CAN)-Part 1: Data Link Layer and Physical Signalling”, https://www.iso.org/standard/63648.html
[2] ISO 11898-1:2015, “Road Vehicles-Controller Area Network (CAN)-Part 2: High-Speed Medium Access Unit”, https://www.iso.org/standard/67244.html
[3] ISO 11898-3:2006, “Road Vehicles-Controller Area Network (CAN)-Part 3: Low-speed, Fault-tolerant, Medium-dependent Interface”, https://www.iso.org/standard/36055.html
[4] T. Kang, J. Lee and S. Lee, “Counterattack Method against Hacked Node in CAN Bus Physical Layer,” *j.inst.Korean.electr.electron.eng.*, vol.23, no.4, pp.1469-1472, 2019.
DOI: 10.7471/ikeee.2019.23.4.1469
[5] J. Lee and S. Lee, “Design and Verification of Automotive CAN Controller,” *j.inst.Korean.electr.electron.eng.*, vol. 21, no.2, pp.162-165, 2017.
DOI: 10.7471/ikeee.2017.21.2.162