라즈베리 파이 기반의 Controller Area Network에서 수집 트래픽 플로우를 재현하는 방법

김민준, 김우철, 이준경, 임 혁

광주과학기술원

minjun01@gist.ac.kr, woocheolkim@gist.ac.kr, jungyeonglee@gist.ac.kr, hlim@gist.ac.kr

Replaying Captured Traffic Flows in Raspberry Pi based Controller Area Network

Minjun Kim, Woocheol Kim, Jungyeong Lee, Hyuk Lim

Gwangju Institute of Science and Technology (GIST)

요 약

CAN (Controller Area Network)은 차량 내 ECU (Electrical Control Unit)와 여러 센서 간의 통신을 위한 네트워크이다. CAN 프로토콜은 데이터 encryption 및 authentication과 같은 보안 기능을 제공하지 않기 때문에 이를 악용하는 공격에 취약하다. CAN 프로토콜의 보안을 강화하기 위한 다양한 연구가 진행되고 있다. 그러나 실제 차량을 대상으로 하는 실험은 안전 및 비용 문제가 발생할 수 있으므로, 실제 차량으로부터 CAN 데이터 트래픽만을 수집하고 독립된 CAN 네트워크를 구성하여 트래픽을 재현하는 실험 환경이 요구된다. 본 논문에서는 차량에서 수집한 CAN 데이터 트래픽을 라즈베리 파이로 구성한 CAN에서 재현하는 테스트베드를 구축한다. 라즈베리 파이 한 개가 CAN frame을 전송할 수 있는 최대 속도는 제한되어 있기 때문에, 실제 차량에서 수십 개의 ECU가 생성한 CAN 데이터를 같은 전송 속도로 재현하기 위해서는 여러 대의 라즈베리 파이가 필요하다. 본 논문은 최소한의 라즈베리 파이를 이용하여 원본 CAN 데이터 트래픽을 수집될 때와 동일한 전송 속도로 재현하는 방법을 제시한다.

Ⅰ. 서 론

CAN (Controller Area Network) 통신은 현재 차량 내 ECU (Electronic Control Unit)와 센서 같은 장치가 서로 통신할 때 사용된다. ECU는 차량 내에서 엔진, 브레이크, 가속 페달, 에어백과 같은 전기 시스템을 제어하는 장치이다. 차량에서 ECU와 센서를 포함한 CAN 노드는 CAN bus에 연결되어있다. CAN은 주소 기반 네트워크가 아닌 메시지 기반 네트워크로써, 하나의 CAN 노드가 특정 노드로 CAN frame을 보내지 않고 CAN bus에 연결된 모든 노드에 브로드캐스트한다 [1]. 표준 CAN frame에는 11bit의 identifier와 최대 8byte의 data field 등이 있다. CAN bus에 연결된 각각의 CAN 노드는 identifier를 기반으로 한 filtering mechanism을 사용하여 필요한 CAN frame만 선택적으로 받는다. 그러나 CAN 프로토콜은 브로드캐스트 통신 방식을 사용하고 encryption/authentication 규약이 존재하지 않아 악의적인 ECU 노드에 의한 통신 방해 공격 및 스푸핑 공격 등에 취약하다 [1], [2], [3].

실제 차량에 CAN 공격 및 보안 알고리즘을 적용하는 경우, 차량 내 여러 기능이 오작동을 일으킬 수 있어 위험부담이 크고 상당한 비용이 요구된다. 본 논문에서는 실제 차량에서 수집한 CAN 통신 데이터를 라즈베리 파이로 재현하는 테스트베드를 구축한다. 실제 차량에서는 수십 개의 ECU와 센서가 통신하므로 차량 내 모든 CAN 노드를 라즈베리 파이가 하나씩 담당할 경우 테스트베드를 구축하는데 큰 비용이 필요하다. 또한, 라즈베리 파이 한 대에는 하나의 CAN module을 장착할 수 있고, CAN module의 CAN frame 전송 속도는 최대 1M bit per second (bps)라는 제약이 있어 하나의 라즈베리 파이로는 실제 CAN 통신 환경을 재현할 수 없다 [4]. 본 논문은 최소한의 라즈베리 파이를 이용하여 수집한 CAN 통신 데이터와 동일한 시간 간격을 가지는 데이터를 구현하는 방법을 보인다. 수집한 실제 차량의 CAN 통신 데이터 중 일부를 라즈베리 파이 테스트베드 환경에서 전송 및 캡처하여 이를 실제 차량 데이터와 비교한다.

그림 1. 실제 차량에서 CAN 통신 데이터 수집을 위한 실험 구성도

No.	Time	Protocol	Length	Identifier	Data
350441	212.558164	CAN	32	STD: 0x0000020a	08 00 20 28 00 00 00 00
350442	212.558410	CAN	32	STD: 0x00000300	11 83 ff 00 1f ff 83 73
350443	212.558652	CAN	32	STD: 0x00000330	00 fa ff 04 01 02 50 7b
350444	212.559786	CAN	32	STD: 0x00000210	00 30 00 00 40 00 7c 05
350445	212.560025	CAN	32	STD: 0x00000212	03 ae 00 00 28 3c 08 03
350446	212.560264	CAN	32	STD: 0x00000214	48 3d 08 3d 08 3d 20 1e
350447	212.560515	CAN	32	STD: 0x00000222	00 00 00 00 00 00 00 00
350448	212.560754	CAN	32	STD: 0x00000308	00 03 c0 00 81 6c 08 0d
350449	212.560987	CAN	32	STD: 0x00000312	08 3d 09 d2 09 d2 08 03
350450	212.561231	CAN	32	STD: 0x00000320	00 53 3f ff 04 ff 04 00
350451	212.561473	CAN	32	STD: 0x000003d2	3b 00 0f 02 06 01 00 00

그림 2. Wireshark를 이용해 차량의 CAN frame을 캡처한 모습

Ⅱ. 차량 CAN 통신 캡처 및 분석

그림 1과 같이 PICAN CAN-Bus Board, 라즈베리 파이, OBD2 to DB9 Cable을 이용하여 실험 기자재로 활용 중인 Tivoli 차량의 CAN 통신 데이터를 수집하였다. 그림 2는 Wireshark를 통해 35만 개 이상의 CAN frame을 캡처한 모습을 보여준다.

수집한 차량의 CAN 데이터 트래픽을 재현하기 위해서, 다음과 같은 사항을 고려하였다. 라즈베리 파이 한 대가 최대 전송 속도로 CAN frame을 보낼 때 CAN frame의 inter-arrival time보다 실제 차량의 CAN frame의 inter-arrival time이 더 짧으므로, 전송한 CAN frame은 원본과 다른 inter-arrival time을 가진다. 라즈베리 파이 한 대에서 일련의 CAN frame을 최대 전송 속도인 1M bps로 단순히 반복 전송할 경우, 각각의

CAN frame은 최대 0.312ms의 inter-arrival time을 가진다. 이는 차량에서 수집한 CAN frame의 timestamp 간격이 0.312ms보다 짧은 경우, CAN 노드 라즈베리 파이 한 대만으로 차량의 데이터 트래픽을 재현하는 것이 불가능하다는 것을 의미한다. 따라서 실제 차량과 비슷한 CAN 통신 환경을 구현하려면 CAN frame의 inter-arrival time을 고려해야 한다.

$$t_{i,d} = T_{i+d} - T_i, 1 \le i \le n-d, \forall n, i, d \in N \quad (1)$$

차량에서 수집한 n개의 CAN frame에 대하여 i번째 frame의 timestamp가 T_i일 때, 어떤 d에 대하여 i번째 frame과 i+d번째 frame 간의 시간 간격 $t_{i,d}$를 식 (1)과 같이 두 frame의 timestamp의 차로 정의한다. 그림 3은 d=1, 2, 3, 4, 5일 때, 식 (1)을 만족하는 $t_{i,d}$의 분포를 각각 누적 분포 함수로 나타낸 것이다. t_{iat}는 라즈베리 파이 한 대에서 최대 전송 속도로 일련의 CAN frame을 단순히 반복 전송하는 상황에서 CAN frame의 최대 inter-arrival time으로써 0.312ms이다. 그림 3에서 $d \ge 2$ 일 때, 0.312ms 미만의 $t_{i,d}$의 비율은 0.5% 미만이다. 자연수 k에 대하여, d개의 라즈베리 파이가 35만 개 CAN frame의 dk-d+1번째, dk-d+2번째, … , dk번째 frame을 차례대로 번갈아 가며 전송하는 경우, 0.312ms보다 큰 $t_{i,d}$를 가지는 CAN frame에 대해 실제 차량에서 수집한 CAN 통신 데이터와 동일한 시간 간격으로 전송할 수 있다. 따라서 $t_{i,d}$의 최솟값이 0.312ms보다 큰 조건을 만족하는 d 중 가장 작은 값을 찾을 경우, 최소한의 라즈베리 파이만을 사용하여 실제 차량과 같은 CAN 통신 상황을 재현하는 테스트베드를 구축할 수 있다. 표 1에서는, d=5일 때 $t_{i,d}$의 최솟값이 0.312ms보다 크므로 라즈베리 파이 5대가 번갈아 가며 CAN frame을 전송하면 수집한 CAN 통신 데이터와 동일한 시간 간격을 갖는 데이터를 만들 수 있음을 보여준다.

Ⅲ. 라즈베리 파이를 이용한 CAN 통신 테스트베드

실험을 위하여, 라즈베리 파이 세 대를 이용하여 실제 차량과 비슷한 CAN 통신 환경을 구축한다. 두 대의 라즈베리 파이가 통신할 때, 또 다른 라즈베리 파이를 모니터 노드로 설정하고 Wireshark로 두 라즈베리 파이 간의 트래픽을 캡처한다. 모니터 노드를 제외한 두 라즈베리 파이는 차량에서 수집한 원본 CAN frame 중 맨 앞 750개의 CAN frame을 각각 홀수 번째와 짝수 번째로 나누어 원본과 같은 시간 간격으로 전송한다. 이때 각각의 라즈베리 파이의 첫 frame의 전송 시작 시각은 trigger frame을 통해 동기화한다. 홀수 번째 CAN frame을 전송하는 라즈베리 파이가 CAN bus에 trigger frame을 보내고 특정 시간만큼 sleep 한다. 짝수 번째 CAN frame을 전송하는 라즈베리 파이는 trigger frame을 수신한 후, 특정 시간에서 network delay를 뺀 시간만큼 sleep 한다. 결과적으로 trigger frame을 통해 두 라즈베리 파이의 시간 동기화를 마친 뒤, 두 라즈베리 파이는 실제 차량에서 수집한 CAN 통신 데이터와 동일한 시간 간격으로 CAN frame을 번갈아 가며 전송한다.

두 라즈베리 파이가 CAN bus에 실시간으로 보내는 750개의 CAN frame을 모니터 노드에서 Wireshark로 캡처한 결과, 750개의 CAN frame 중 730개의 frame이 차량에서 수집한 CAN frame과 순서가 같았다. 750개의 CAN frame에 대해서, 첫 번째로 도착한 CAN frame과의 시간 차를 각각 구하였다. 마찬가지로 차량에서 미리 수집한 CAN frame 중 750개의 CAN frame에 대해서도 첫 CAN frame과의 시간 차를 구하였다. 최종적으로 두 결과를 비교하였고, 그 오차는 평균 0.07094ms이다.

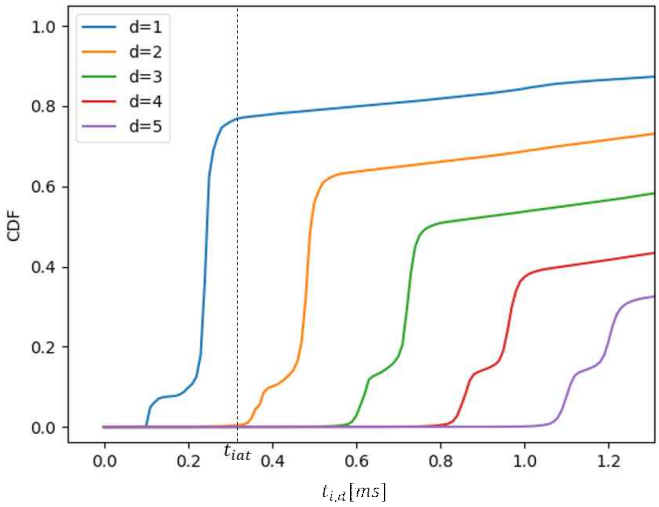

그림 3. 차량에서 수집한 35만 개의 CAN frame의 $t_{i,d}$의 누적 분포 함수

표 1. 차량에서 수집한 35만 개의 CAN frame의 $t_{i,d}$의 최솟값

d	minimum value of $t_{i,d}$ [ms]
1	0.001907
2	0.005960
3	0.1340
4	0.2439
5	0.3791

Ⅳ. 결론

본 논문에서는 실제 차량에서 직접 CAN 데이터 트래픽을 수집하고, 해당 CAN 데이터 트래픽을 최소한의 라즈베리 파이로 구성된 테스트베드에서 재현하는 방법을 보인다. 실험을 통하여 실제 차량의 CAN 통신 데이터 일부를 라즈베리 파이 테스트베드 환경에서 전송 및 캡처하고 이를 실제 차량 데이터와 비교하였다.

ACKNOWLEDGMENT

이 성과는 2021년도 광주과학기술원의 지원을 받아 수행된 연구임 (K14120, 인공지능 서비스의 안정성과 보안성 연구).

참 고 문 헌

[1] R. Buttigieg, M. Farrugia and C. Meli, “Security Issues in Controller Area Networks in Automobiles,” International Conference on Sciences and Techniques of Automatic Control and Computer Engineering, pp. 93-98, 2017.

[2] C. Lin and A. Sangiovanni-Vincentelli, “Cyber-Security for the Controller Area Network (CAN) Communication Protocol,” International Conference on Cyber Security, pp. 1-7, 2012.

[3] H. Ueda, R. Kurachi, H. Takada, T. Mizutani, M. Inoue, and S. Horihata, “Security Authentication System for In-Vehicle Network,”, SEI Technical Review, no. 81, pp. 5 - 9, 2015.

[4] H. Chen and J. Tian, “Research on the Controller Area Network,” International Conference on Networking and Digital Society, pp. 251-254, 2009.