

Automotive Cybersecurity Threats and Countermeasures

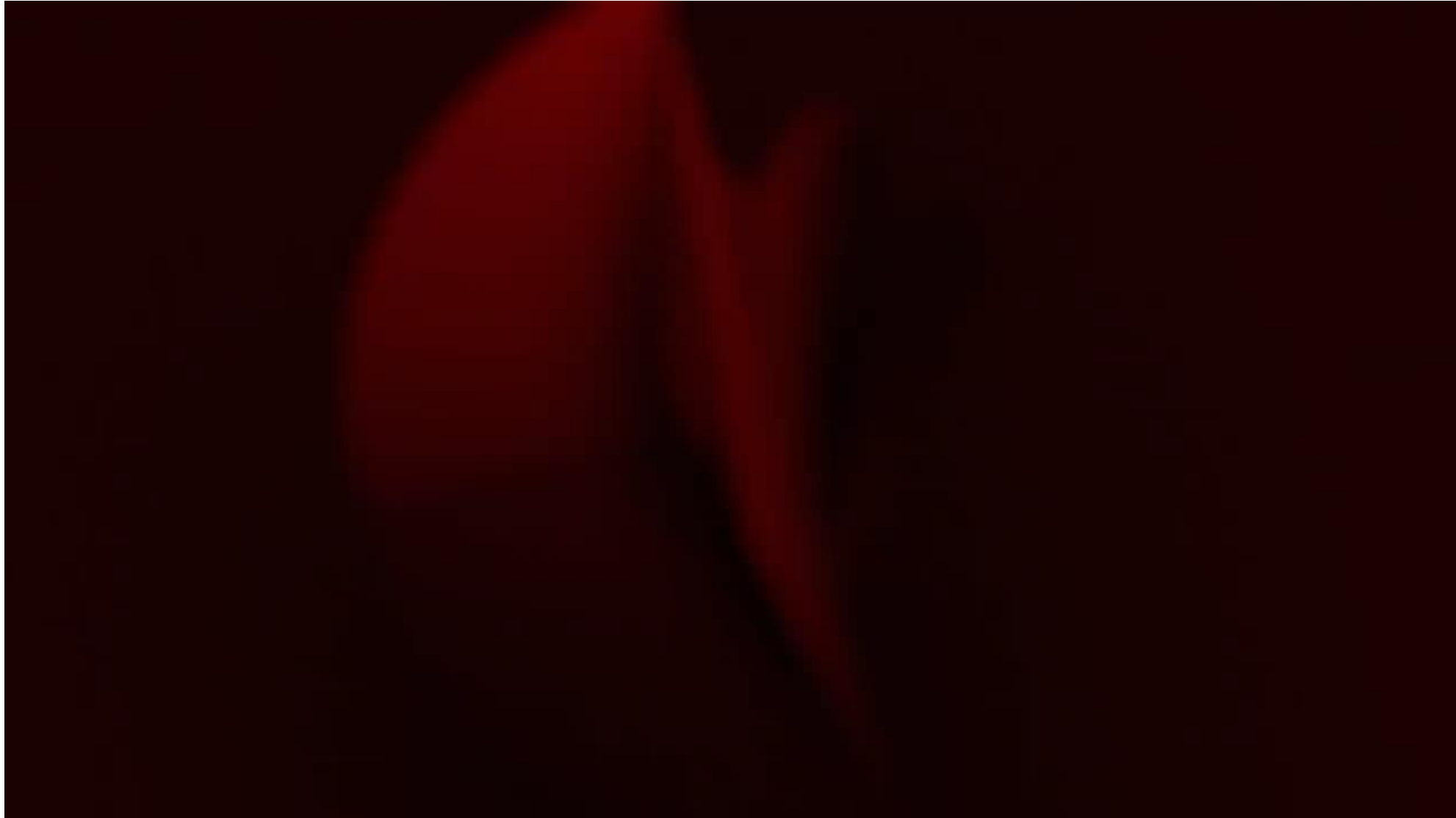
2024.07.07

국민대학교

전상훈 교수

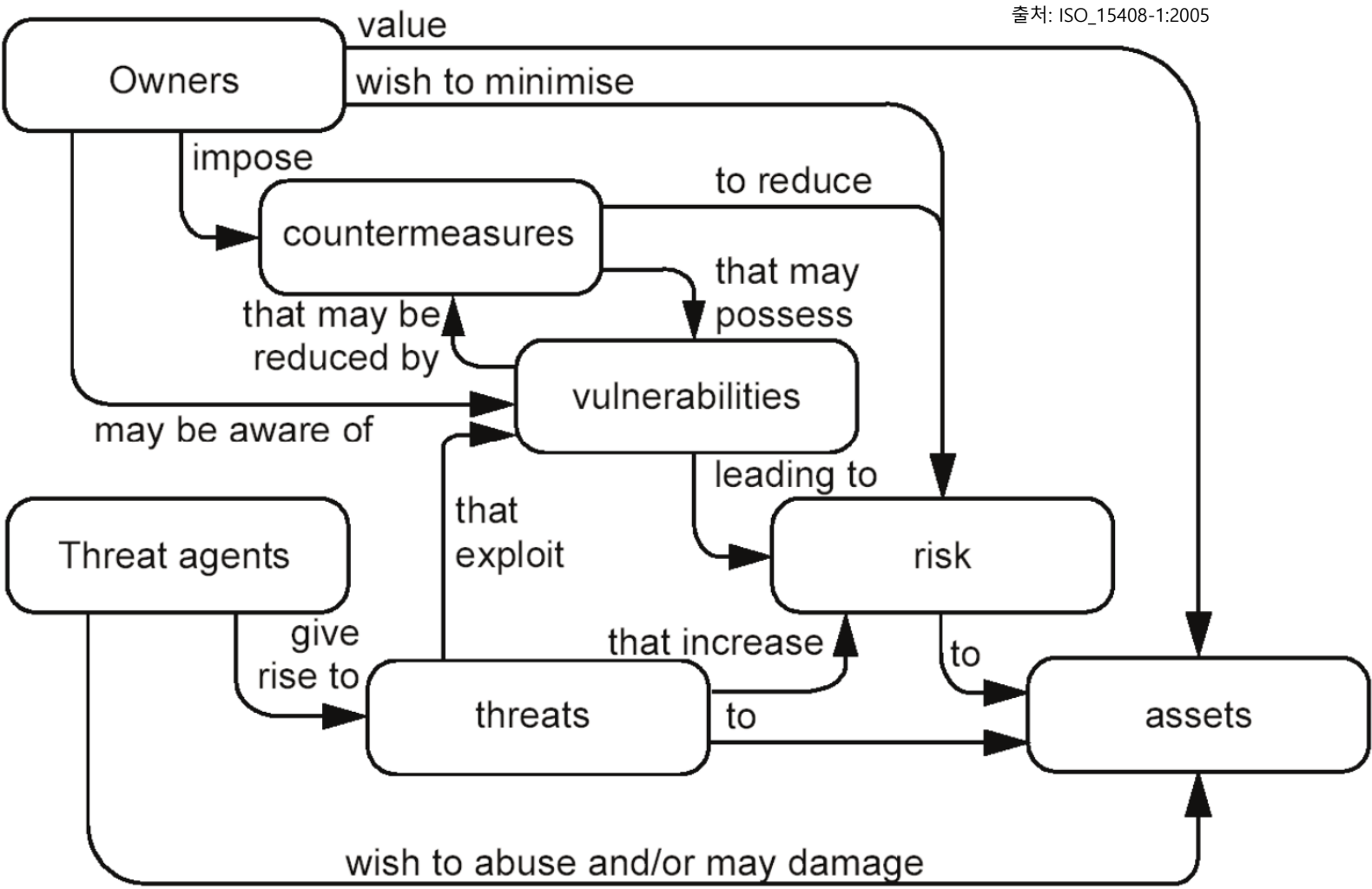
CONTENTS

- 01** 자동차 사이버보안 개요
- 02** 자동차 사이버공격 사례
- 03** 자동차 사이버공격 대응기술



01

자동차 사이버보안 개요



[그림] Security Concepts and Relationships



이미지 출처: Adobe Stock

■ Spoofing

- 사전적인 의미로 '속이다' 라는 뜻이다. 즉, Spoofing 공격이란 속이는 것을 이용한 공격 기법들을 의미한다.
- ARP (Address Resolution Protocol, IP → MAC) Spoofing
 - 중간자 공격 (Man-In-the-Middle Attack)
- E-mail Spoofing
 - 보통 Spam Mail이라는 용어로 사용

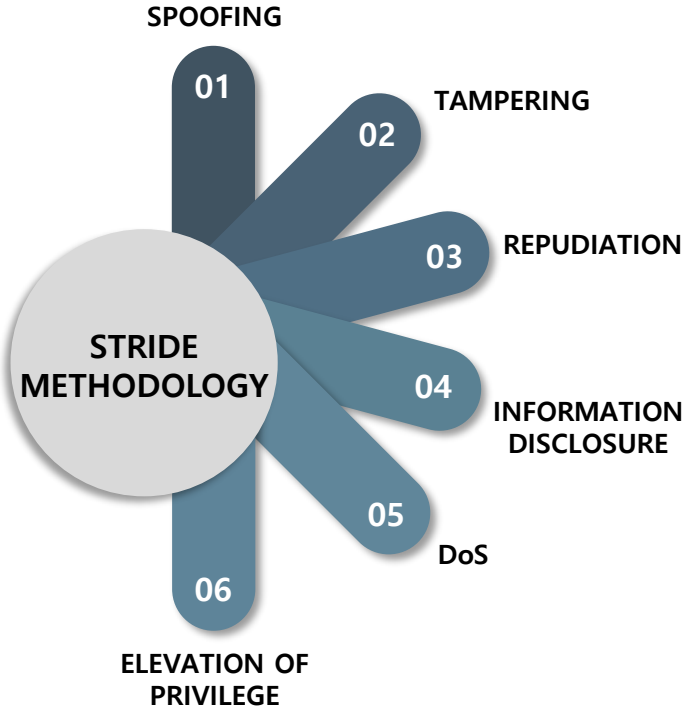
■ Sniffing (Snooping)

- 네트워크 상에서 자신이 아닌 다른 상대방들의 패킷 교환을 엿듣는 것
- Eavesdropping, Wiretapping, Hijacking 등과 유사한 의미로 사용
 - 단, hacking 목적이 다를 수 있음



이미지 출처: Adobe Stock

	Threat	Security Attribute	Threat Definition
S	Spoofing identity	Authentication	Pretending to be something or someone other than yourself
T	Tampering with data	Integrity*	Modifying something on disk, network, memory, or elsewhere
R	Repudiation	Non-repudiation	Claiming that you didn't do something or were not responsible; can be honest or false
I	Information Disclosure	Credential*	Providing information to someone not authorized to access it
D	Denial of Service (DoS)	Availability*	Exhausting resources needed to provide service
E	Elevation of Privilege	Authorization	Allowing someone to do something they are not authorized to do



*Security 3대 요소

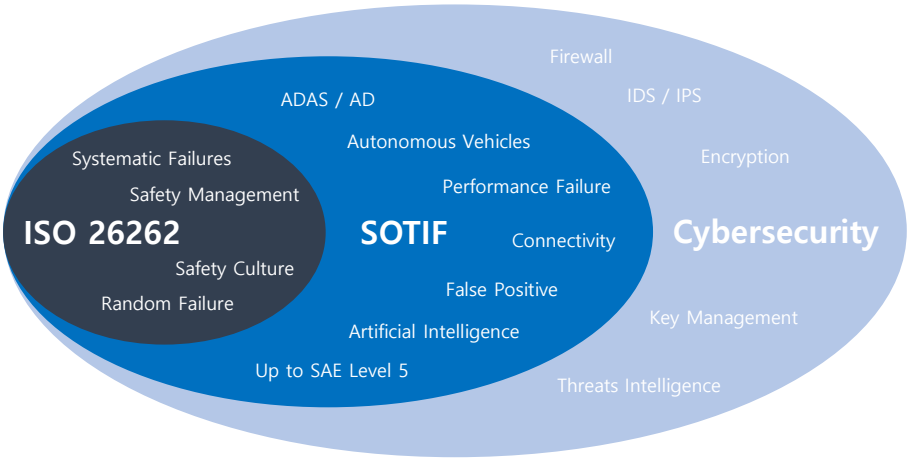
	위험도	공격에 따른 결과 예측
D	Damage potential	피해는 얼마나 클 것인가?
R	Reproducibility	공격이 성공할 확률이 얼마인가?
E	Exploitability	공격을 위해 얼마나 많은 노력과 기술이 필요한가?
A	Affected users	위험으로부터 얼마나 많은 사용자가 영향을 받는가?
D	Discoverability	얼마나 쉽게 취약성이 발견되는가?

- 자동차는 IT 기술을 적극적으로 사용하여 운전의 편의성과 안전성을 높여 왔음. 예를 들어, 운전자 보조 기술, 인포테인먼트, 차량 간 통신(V2V), 차량-인프라 통신(V2I) 등의 기술이 대중화되면서 운전의 편리성은 증가하지만, 이에 따른 보안 위협 상승
- 자동차 사이버보안(Automotive Cybersecurity)은 자동차 내부 시스템과 통신 네트워크를 보호하여 자동차를 사이버 공격으로부터 안전하게 유지하는 기술
 - 즉, 최신 자동차에 사용되는 많은 전자 기술과 통신 기술을 보호하여 해킹, 바이러스, 악성 코드 등으로 인한 위협으로부터 차량을 보호하고 안전한 운행을 보장하는 기술
 - 자동차에 대한 사이버 공격은 운전자와 승객의 생명과 안전을 직접적으로 위협함. 그리고 이러한 공격은 자동차 내부의 기술적 결함이나, 인프라 시스템의 취약점 등을 통해 이루어질 수 있음



이미지 출처: Adobe Stock

	ISO26262	ISO21448 (SOTIF)	ISO21434 (CSMS)
개요	- 기능 안전 (Functional Safety) - 차량에 탑재되는 E/E (Electric/Electronic) 시스템 결함	- 의도된 기능으로부터의 안전 - E/E 시스템 결함이 없는 기능에서 발생할 수 있는 위험	- 외부 위협으로부터의 안전 - 차량 생산 생명주기를 기준으로 사이버보안 분야에 대한 주요 접근 방법론 제시
내용	정의된 기능(요구사항)을 충실히 구현하였는지 여부를 검증	정의된 기능 자체가 의도치 않은 결함을 발생시킬 수 있는지 여부를 검증	의도성이 있는 외부 위협으로부터 안전한지 여부를 검증
중점	Safety (안전)		Security (보안)



[그림] ISO26262, SOTIF, Cybersecurity의 내용과 범위

■ UNECE WP.29에서 의무적으로 적용(22년 7월) 해야 하는 사이버 보안 규정 (UNR.155) 채택

- CSMS* 지원을 위해 차량 라이프 사이클 전반에 걸친 사이버 보안 활동 관리 필요 (TARA: Threat Analysis and Risk Assessment)

■ 시스템 오동작, Sensor Spoofing, 차량 해킹 등 차량 안전 및 보안 관련 위협 증가

- 커넥티드 카, 자율주행 차량 외 상당수의 자율주행체(UAV 등)가 유사한 문제점을 가짐



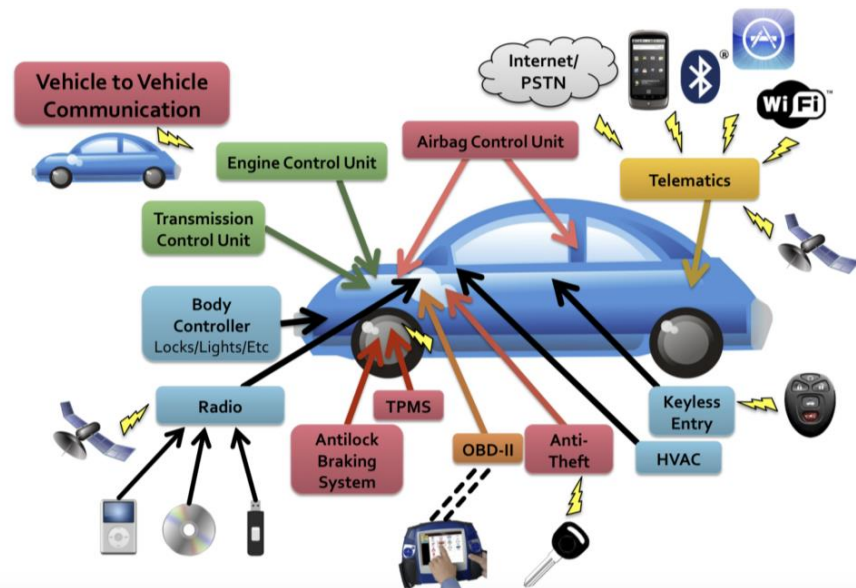
시스템 오동작으로 인한 사고사례
(Tesla Autopilot, Uber, Google Waymo)



드론을 이용한 테러 위협
(출처 : MBC News, TV조선)

■ 보안 결함으로 인한 차량 리콜 증가

- C사 (원격제어, 2015년/2016년)
- H사 (커넥티드서비스, 2017년 / 이모빌라이저, 2022년)
- T사 (원격제어, 2016년/2017년 / 후미등, 2022년/2023년)



[그림] 자동차 내부 네트워크에 접근 가능한 Attack Surface

출처: Checkoway, Stephen, et al. "Comprehensive experimental analyses of automotive attack surfaces." USENIX Security Symposium. Vol. 4. 2011.

■ C.A.S.E

- 미래자동차는 연결(Connectivity), 자율주행(Autonomous), 공유(Sharing), 전동화(Electrification)라는 4가지 키워드로 정리할 수 있다.



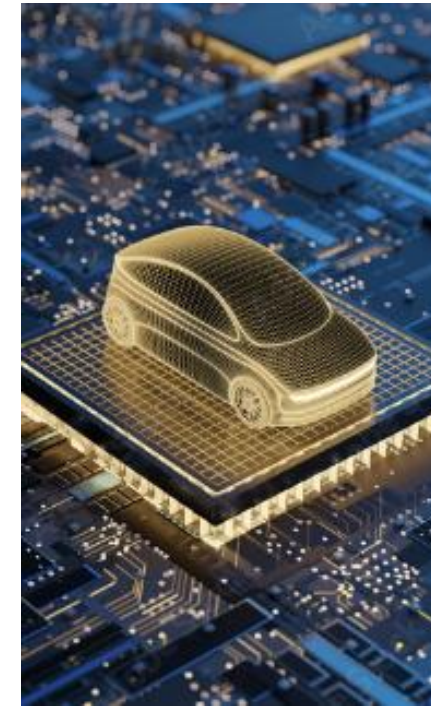
Connectivity



Autonomous



Sharing



Electrification

이미지 출처: Adobe Stock

■ C.A.S.E

- 미래자동차는 연결(Connectivity), 자율주행(Autonomous), 공유(Sharing), 전동화(Electrification)라는 4가지 키워드로 정리할 수 있다.



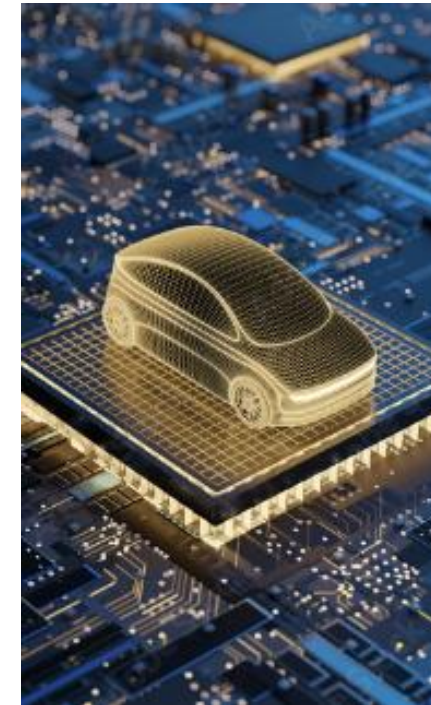
Connectivity



Autonomous



Security

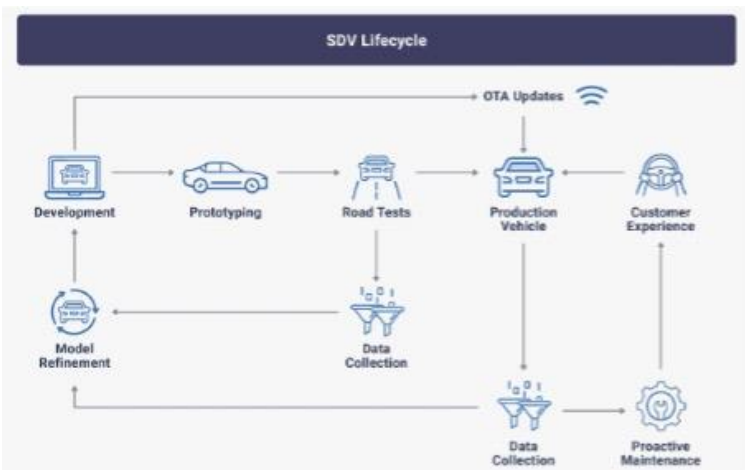


Electrification

이미지 출처: Adobe Stock

SDV 기술의 정의 및 의미

- ❖ SDV란 제품 수명주기 동안 OTA를 통해 성능/기능을 개선 및 추가, 다양한 서비스와 콘텐츠를 제공하여,
 - “고객”에게 지속적으로 새로운 경험의 가치를 줄 수 있는 차량,
 - “OEM”에게 개발 자원 절약 및 데이터 활용으로 제품 수명주기동안 지속적인 수익을 창출 하는 차량
 - “부품사”에게 HW규모의 경제와 SW의 부가가치를 창출하는 차량,
 - “IT/SP*”에게 APP 또는 플랫폼 생태계를 확장할 수 있는 차량
- ❖ 자동차의 수명 주기가 연장되고 SOP** 이후에도 성능 향상 가능



SDV Lifecycle (출처: BlackBerry QNX)

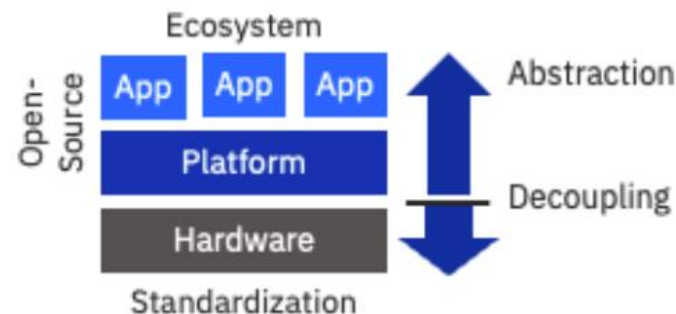
SDV 기술의 BASE

SP*: Service Provider
SOP**: Start of Production

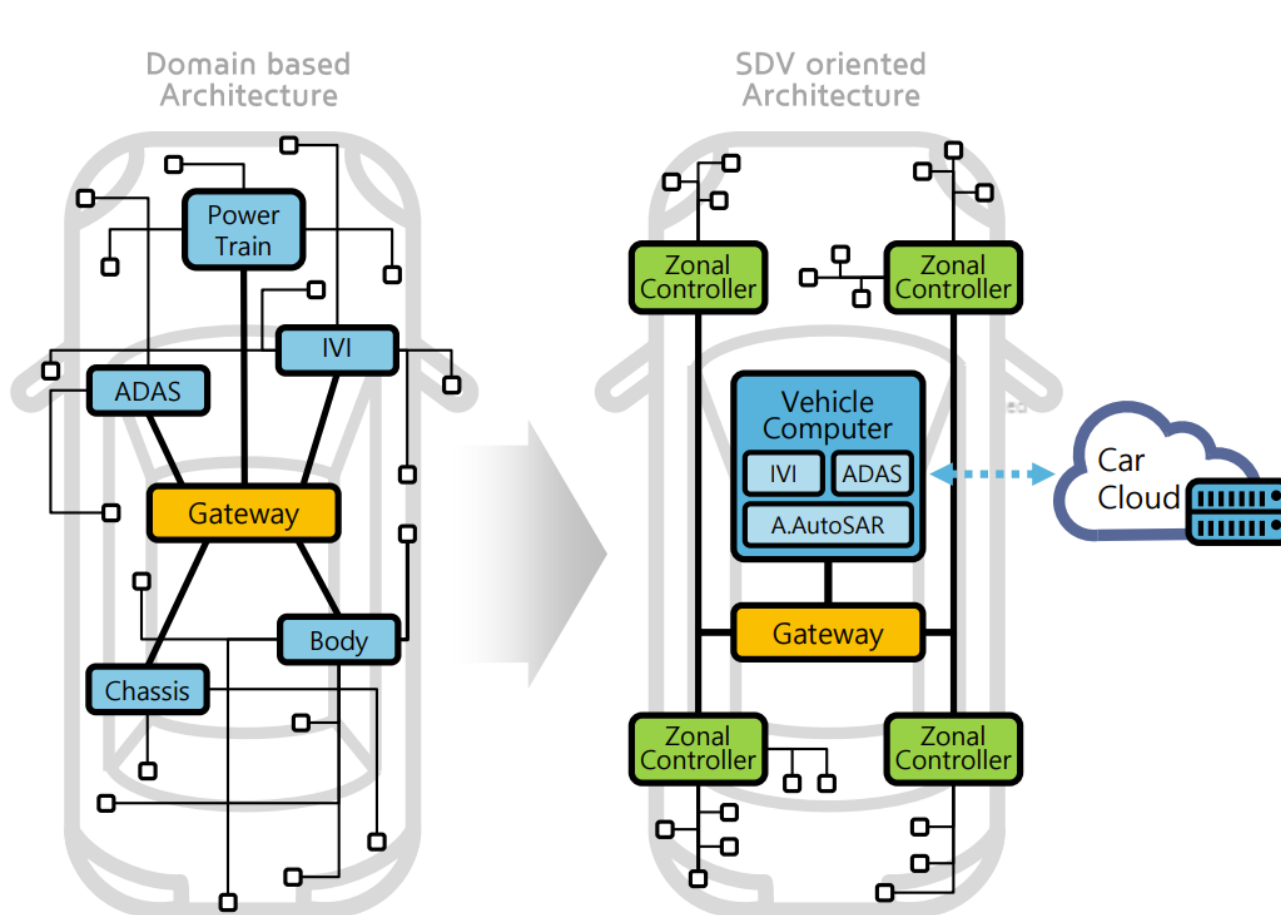
- ❖ SDV 기술의 기본(Base) 정의: 구성 요소
 - ECU 통합화 및 중앙화 E/E Architecture 기반으로 차량 복잡도와 비용 감소
 - 하드웨어 - 소프트웨어 디커플링을 통한 새로운 서비스/기능 제공
 - 소프트웨어 플랫폼 / 클라우드 서비스로 차량 기능 확장, 다양화
 - 차량 소프트웨어 OTA 업데이트로 차량 수명 주기와 성능 증대



- ❖ 디커플링, 추상화, 표준화 및 오픈소스를 통해 앱생태계 생성



SW 정의에 따라 차량의 특성이 정의 가능한 아키텍처 확보 필요



'도메인 중심의 Architecture'의 한계

- 차량 내 기능 증가에 따라 컨트롤 장치 (DCU) 증가
- 차량 내 연결 배선의 증가로 복잡성/중량 증가
- DCU-부품 간 연동을 위한 HW/SW 복잡성 증가
- 기능 개선 향상을 위해 많은 비용과 시간 필요
- DCU 기반의 Architecture 유연성에 한계

'SDV 지향 Architecture'로의 진화

- 여러 개의 도메인 컨트롤러를 하나의 고성능 제어기로 대체
- 배선의 단순화, 개발 비용과 차량 무게의 축소
- VC-Cloud 간 통신을 통한 용이한 OTA 업데이트
- HW의 추가보다 SW정의에 의해 차량의 특성 결정

02

자동차
사이버공격 사례

OBD*: On-Board Diagnostics
CAN*: Controlled Area Network

■ Experimental Security Analysis of a Modern Automobile (by University of Washington)

- 세계 최초의 자동차 사이버보안 위협에 대한 연구
 - ECU가 본격적으로 차량에 탑재하기 시작한 초기인 점을 감안하면 연구 결과가 매우 시기 적절하였음
 - 현재 자동차 사이버보안 기술 연구의 초석으로 평가받음
- 자동차 진단을 위한 OBD** 표준 인터페이스를 통해 CAN** 통신 Trace 분석 후 제어 메시지 주입

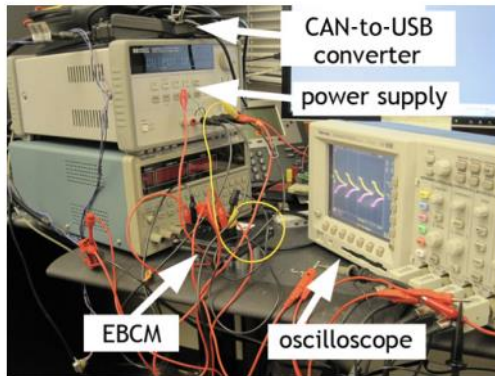


Figure 1. Example bench setup within our lab. The Electronic Brake Control Module (EBCM) is hooked up to a power supply, a CAN-to-USB converter, and an oscilloscope.

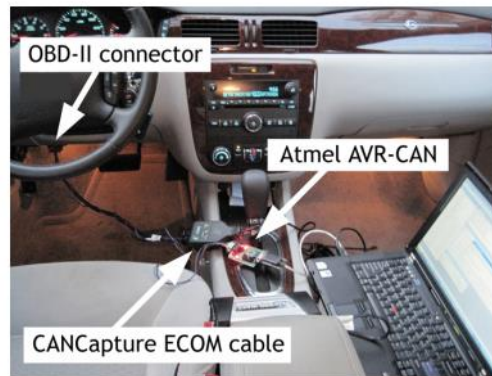


Figure 2. Example experimental setup. The laptop is running our custom CARSHARK CAN network analyzer and attack tool. The laptop is connected to the car's OBD-II port.



Figure 3. To test ECU behavior in a controlled environment, we immobilized the car on jack stands while mounting attacks.

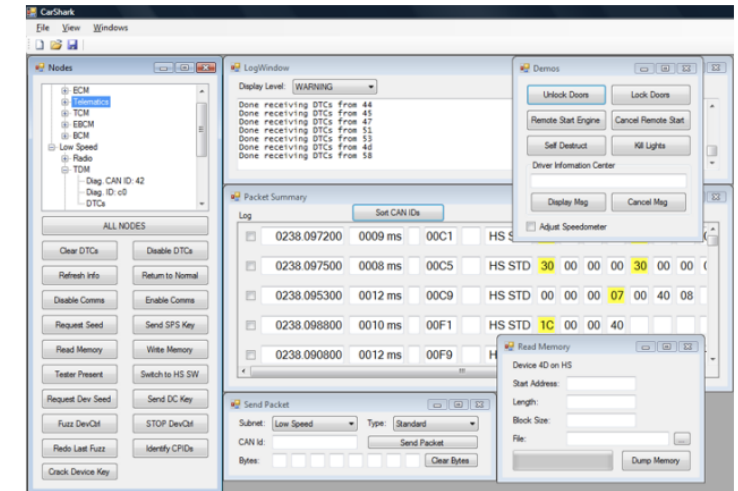
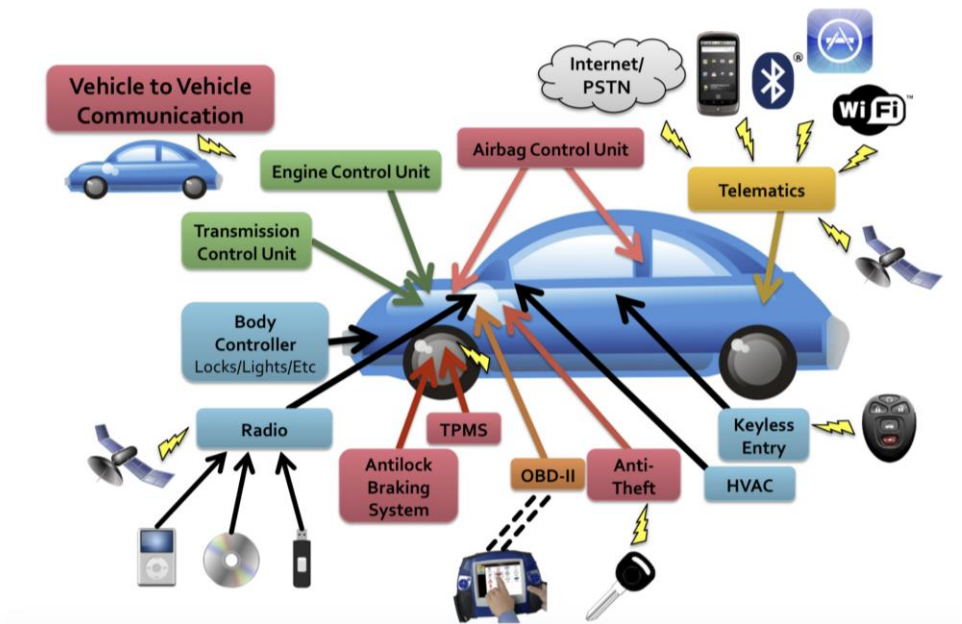


Figure 4. Screenshot of the CARSHARK interface. CARSHARK is being used to sniff the CAN bus. Values that have been recently updated are in yellow. The left panel lists all recognized nodes on high and low speed subnets of the CAN bus and has some action buttons. The demo panel on the right provides some proof-of-concept demos.

■ Comprehensive experimental analysis of automotive attack surfaces (by University of California)

- 이전 연구의 후속 연구로 외부에서 자동차 내부 네트워크에 접근 가능한 Attack Surface에 대한 연구 진행
- 각 Attack Surface에 대한 공격 시나리오 생성
 - Aqlink 프로토콜에서 허용되는 패킷의 최대 크기가 1024 bytes, 그러나 텔레매틱스 장비에서 해당 패킷을 처리하는 버퍼는 100 byte임 → buffer overflow 버그로 연결됨
 - 인증과정에서 사용되는 random number generator의 seed가 장비 off 과정에서 초기화 되는 flaw 발견



Vulnerability Class	Channel	Implemented Capability	Visible to User	Scale	Full Control	Cost
Direct physical	OBD-II port	Plug attack hardware directly into car OBD-II port	Yes	Small	Yes	Low
Indirect physical	CD	CD-based firmware update	Yes	Small	Yes	Medium
	CD	Special song (WMA)	Yes*	Medium	Yes	Medium-High
	PassThru	WiFi or wired control connection to advertised PassThru devices	No	Small	Yes	Low
Short-range wireless	PassThru	WiFi or wired shell injection	No	Viral	Yes	Low
	Bluetooth	Buffer overflow with paired Android phone and Trojan app	No	Large	Yes	Low-Medium
	Bluetooth	Sniff MAC address, brute force PIN, buffer overflow	No	Small	Yes	Low-Medium
Long-range wireless	Cellular	Call car, authentication exploit, buffer overflow (using laptop)	No	Large	Yes	Medium-High
	Cellular	Call car, authentication exploit, buffer overflow (using iPod with exploit audio file, earphones, and a telephone)	No	Large	Yes	Medium-High

[Ref] Checkoway, Stephen, et al. "Comprehensive experimental analyses of automotive attack surfaces." 20th USENIX security symposium (USENIX Security 11). 2011.

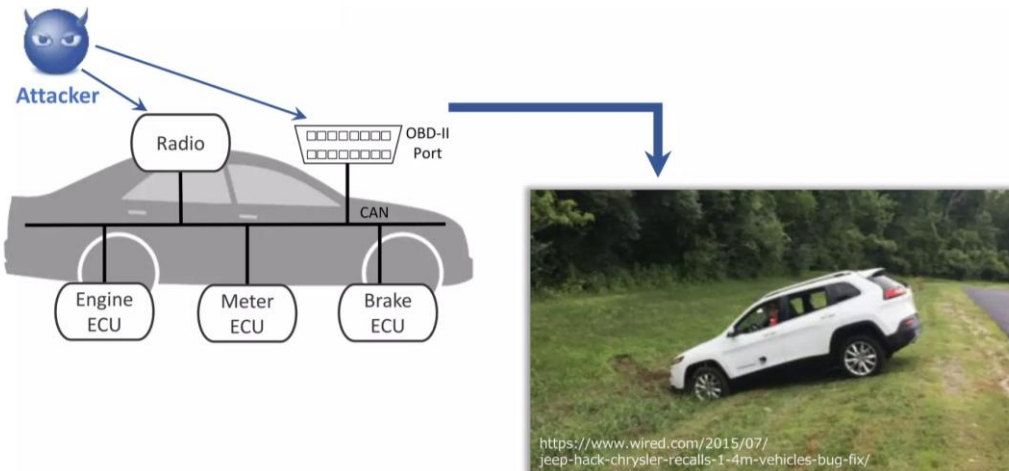
[Ref] Miller, Charlie, and Chris Valasek. "Adventures in automotive networks and control units." Def Con 21.260-264 (2013): 15-31.

■ Adventures in automotive networks and control units ('13)

- Miller and Valasek은 앞선 Koscher et al.의 연구('10)를 재연하여 2013년 Def Con21에 발표
 - 이전 연구의 경우 실험한 차량의 정보나 주입한 CAN 메시지에 대하여 모두 블라인드 처리한 반면, 본 연구 결과는 실험한 차량의 정보와 주입한 CAN 메시지 모두 공개
 - 특히, 본 연구는 Unified Diagnostic Services 진단 표준 프로토콜을 활용하여, 표준이 적용된 모든 차량에 적용할 수 있음

■ Remote exploitation of an unaltered passenger vehicle ('15)

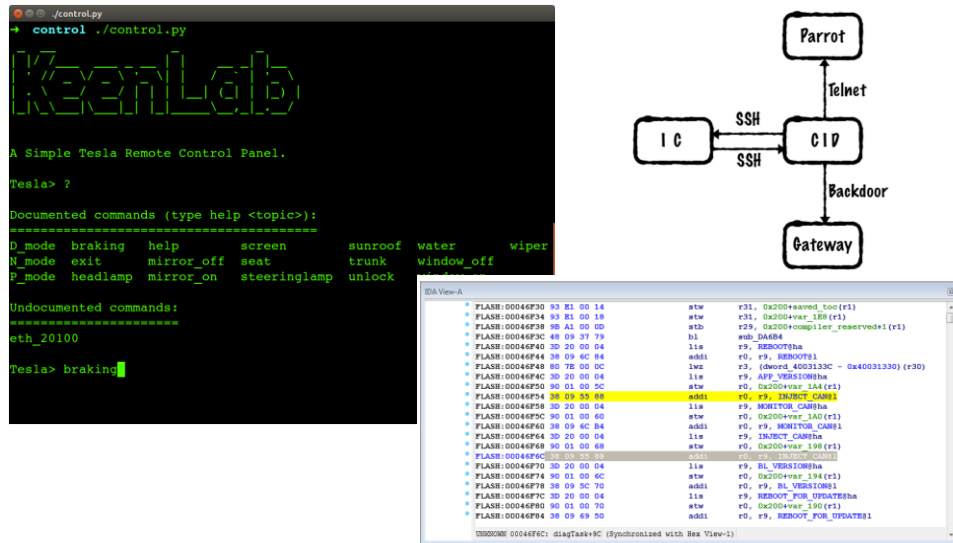
- 기존 연구는 자동차 내부 네트워크에 접근을 한 상태에서 CAN 메시지를 주입함으로써 차량을 제어
- 본 연구는 사전 조작이 되지 않은 차량을 대상으로 원격에서 자동차 내부 네트워크에 접근하여 차량을 제어할 수 있음을 최초로 보임



[Ref] Miller, Charlie, and Chris Valasek. "Remote exploitation of an unaltered passenger vehicle." Black Hat USA 2015.S 91 (2015): 1-91.

■ Free-Fall: Hacking Tesla from wireless to CAN bus

- 중국의 Tencent의 Keen 보안 연구팀이 Tesla를 대상으로 원격 공격 성공
 - 무선(Wi-Fi/Cellular)을 통해 침입하여 IC(Infotainment Cluster), CID(Center Information Display), GW(Gateway)와 같은 차량 내 여러 시스템을 손상
 - 악성 CAN 메시지를 CAN Bus에 주입하여 공격
- Tesla에 연구 결과를 제출한 지 열흘만에 Tesla는 OTA 메커니즘을 사용한 업데이트를 통해 대응
 - 코드 서명 보호 기능을 Tesla에 도입함



JULY 25-27, 2017
MANDALAY BAY / LAS VEGAS, NV

Who we are & What we did

Tencent

- Keen Security Lab of Tencent. (aka KeenLab or KeenTeam)
- Researchers in KeenLab who are focusing on the cutting-edge security research of smart cars.
- In September 2016, successfully implemented a remote attack on the Tesla Model S in both Parking and Driving mode. This remote attack utilized a complex chain of vulnerabilities.
- More Information
 - <https://www.youtube.com/watch?v=c1XyhReNcHY>
 - <http://keenlab.tencent.com/en/2016/09/19/Keen-Security-Lab-of-Tencent-Car-Hacking-Research-Remote-Attack-to-Tesla-Cars/>
 - <https://www.wired.com/2016/09/tesla-responds-chinese-hack-major-security-upgrade/>

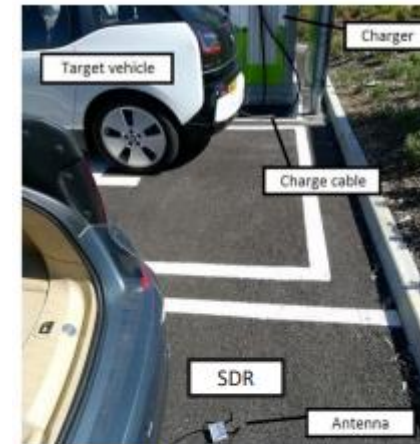
[Ref] Nie, Sen, Ling Liu, and Yuefeng Du. "Free-fall: Hacking tesla from wireless to can bus." Briefing, Black Hat USA 25.1 (2017): 16.

■ (Passive attack) 충전 케이블에서 발생하는 EM(Electromagnetic) leakage를 악용한 도청 공격

- SDR(Software defined Radio)를 이용한 도청장치 구현
- 다양한 충전소 환경에서 도청 공격을 수행하여 통신 패킷 복원
- 차량의 식별자 정보(MAC)가 평문으로 전송되고 있음을 분석함

Site	Antenna	Peak SNR (dB)	BW (MHz)	Total PPDUs	Data PPDUs	Bi-direc.?	Start?	RX% Mean	Min	CRC32% Mean	Max
A	In car	15	6	526	272	✓		99.3	1.1	1.8	3.3
B	In car	18	12	1063	567	✓		29.8	0.5	3.3	5.3
C	In car	25	14	2976	1819	✓		99.9	46.6	48.1	50.3
D	In car	10	12	556	293	✓		88.2	1.4	2.3	3.0
E	In car	9	4.5	569	306			100	11.0	11.1	11.2
F	In car	21	12	3660	2009	✓	✓	99.3	27.8	36.8	45.8
	Bay behind	15	8	1434	1430	✓		99.3	43.5	43.5	43.5
	Outside car	10	10	12987	8255	✓		76.2	34.9	46.6	89.5
G	Two cars	14	11	2449	2274			99.1	24.3	47.5	70.8
	In car	19	12	5837	3670	✓	✓	99.0	51.1	60.3	71.4
	Next bay	15	13	4157	2749	✓		99.7	91.8	91.8	91.8
H	By cable	29	23	23984	17246	✓	✓	80.2	52.9	74.0	99.8
	In car	16	12.5	15052	9362	✓		99.2	69.9	71.0	72.8
	Outside car	20	11	16243	10407	✓		99.5	27.7	61.6	80.6
I	By cable	35	25	19535	14717	✓	✓	92.1	34.2	70.0	92.8
	Two cars	15	12	24121	21006			99.6	42.2	71.9	94.8
	In car	20	12	1501	1193	✓	✓	98.0	94.8	97.4	100.0
J	In car	20	7	14231	10291	✓	✓	81.0	1.0	33.6	67.9
	Outside car	23	7	1084	935	✓	✓	96.0	49.2	49.2	49.2
	In car	8	5	1971	1278	✓		92.5	0.0	22.0	38.3
L	Outside car	8	7	3004	1849		✓	25.8	0.0	0.0	0.0
M	In car	20	12	13631	9743	✓	✓	98.8	42.4	64.9	82.5
N	In car	24	14	4317	3364	✓	✓	68.3	0.0	44.5	72.6

<다양한 장소에서의 공격 성능 평가>



<실험 환경 구성>

Vehicle	MAC
BMW i3	f0:7f:0c:02:●:●
VW e-Golf	00:7d:fa:01:●:●
Jaguar I-PACE	00:1a:37:70:●:●

<복원된 차량 고유 식별자 정보>

■ TBONE – A zero-click exploit for Tesla MCUs

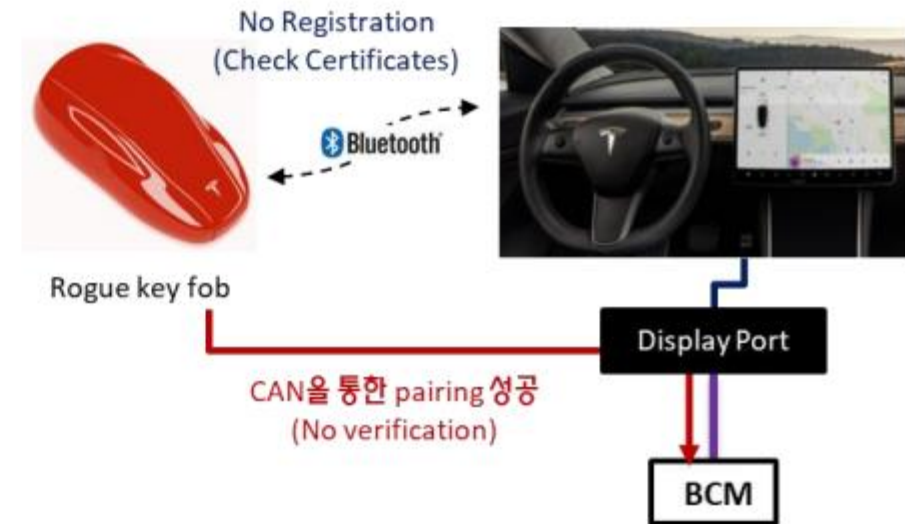
- Tesla 모델이 사용하고 있는 WiFi 관련 오픈소스에 취약점이 존재함을 발견함
 - Tesla 모델은 WiFi 관련 오픈소스인 ConnMan (무선망 연결관리)를 사용하고 있는 것으로 파악됨
 - 해당 오픈소스의 one-day 취약점을 이용하여 차량의 문 열기, 좌석 위치 변경 등을 수행할 수 있음을 보임
 - Initial entry over WiFi
 - ✓ Tesla's SSID
 - Tesla의 Autoshop 서비스 센터는 "Service WiFi"라는 AP 서비스를 제공함
 - 해당 무선 채널은 WPA2-PSK 보안 설정이 되어 있음
 - 하지만 해당 무선 채널 보안에 사용되는 password는 SNS 등에서 쉽게 찾을 수 있음
 - ✓ ConnMan
 - ConnMan은 효율적인 무선망 관리를 지원함
 - Fuzzing with AFL fuzzer (forward_dns_reply() function in src/dnsproxy.c) → uncompress() function (메모리 버그 발견)



```
Reading symbols from src/connmand...done.
Starting program: ./src/connmand < out/crashes/id:000003,sig:06,src:006692,op:havoc,rep:8
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
=====
==28665==ERROR: AddressSanitizer: negative-size-param: (size=-1)
    #0 0xf720f9b3 (/usr/lib32/libasan.so.4+0x779b3)
    #1 0x56936544 in uncompress src/dnsproxy.c:1841
    #2 0x5694ba34 in forward_dns_reply src/dnsproxy.c:2086
```

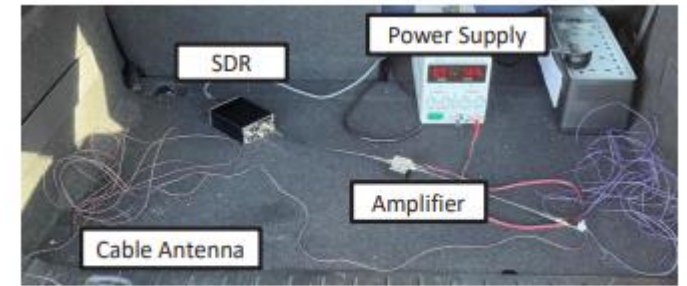
■ Tesla Model X의 Bluetooth 기반 스마트키 공격

- Tesla Model X의 스마트키 취약점이 보고됨
 - Bluetooth 기반 key fob 업데이트 취약점
 - CAN network 기반 key fob pairing 프로토콜 취약점

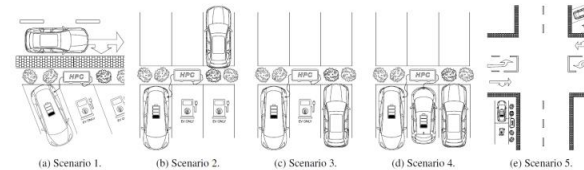


■ Brokenwire: Wireless Disruption of CCS Electric Vehicle Charging

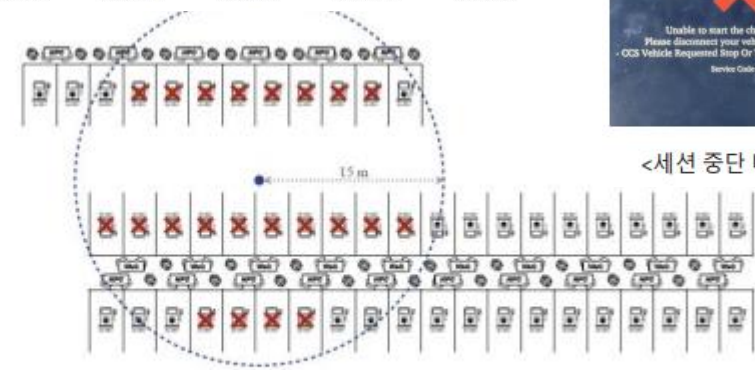
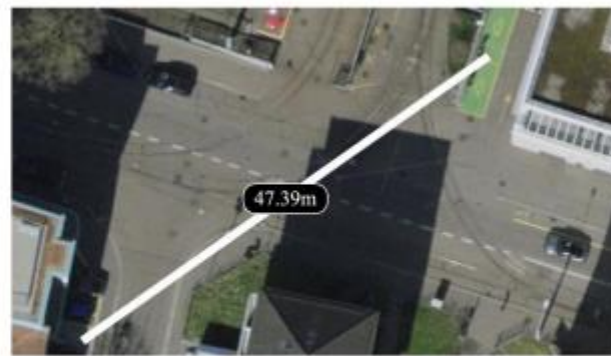
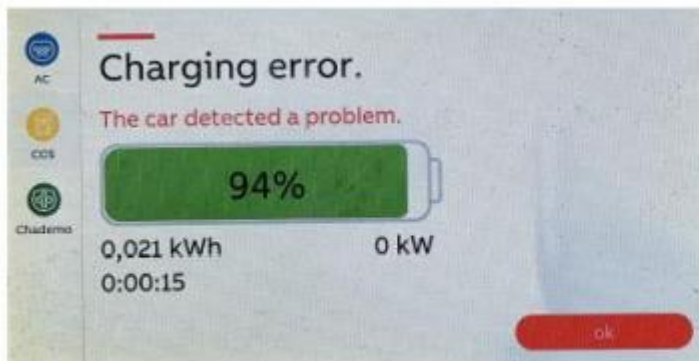
- (Active Attack) 신호 주입 공격을 통해 세션을 중단시키는 DoS 공격
 - PLC 통신 보드 구현을 통한 프로토타입 검증 수행
 - SDR을 이용하여 low-cost 신호 주입 공격 장치를 구현함
 - 야외에서 최대 47m 거리에서 충전 세션을 중단시킬 수 있음을 확인함
 - 실내의 경우 다른 층(floor)에 위치한 공격자에 의해서도 공격이 가능함을 확인
 - 현재 Responsible Disclosure로 인해 세부 공격은 확인하기 어려움



<신호 주입 공격 모듈 구성>



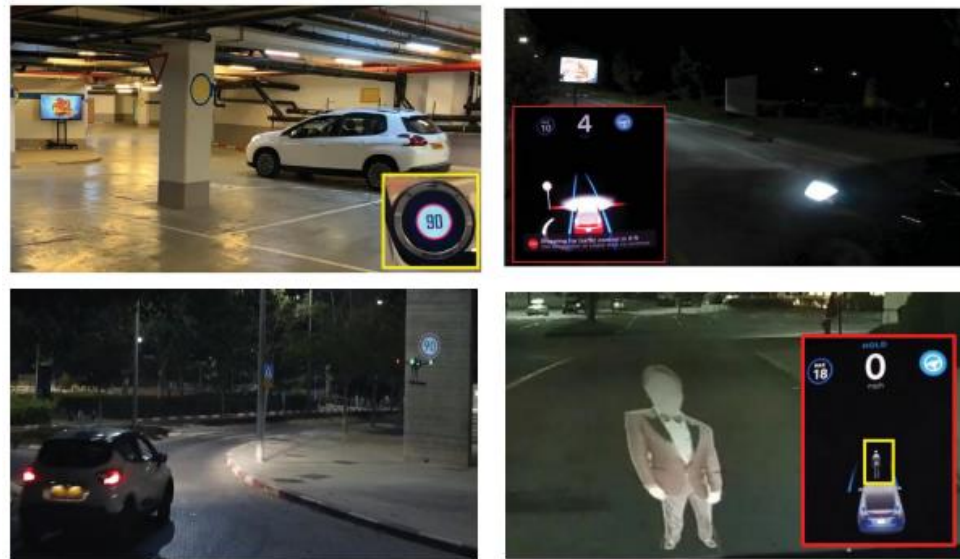
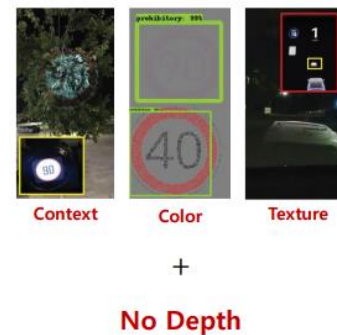
<세션 중단 메시지>



15m반경에서 22대의 차량을 동시에 공격 가능함

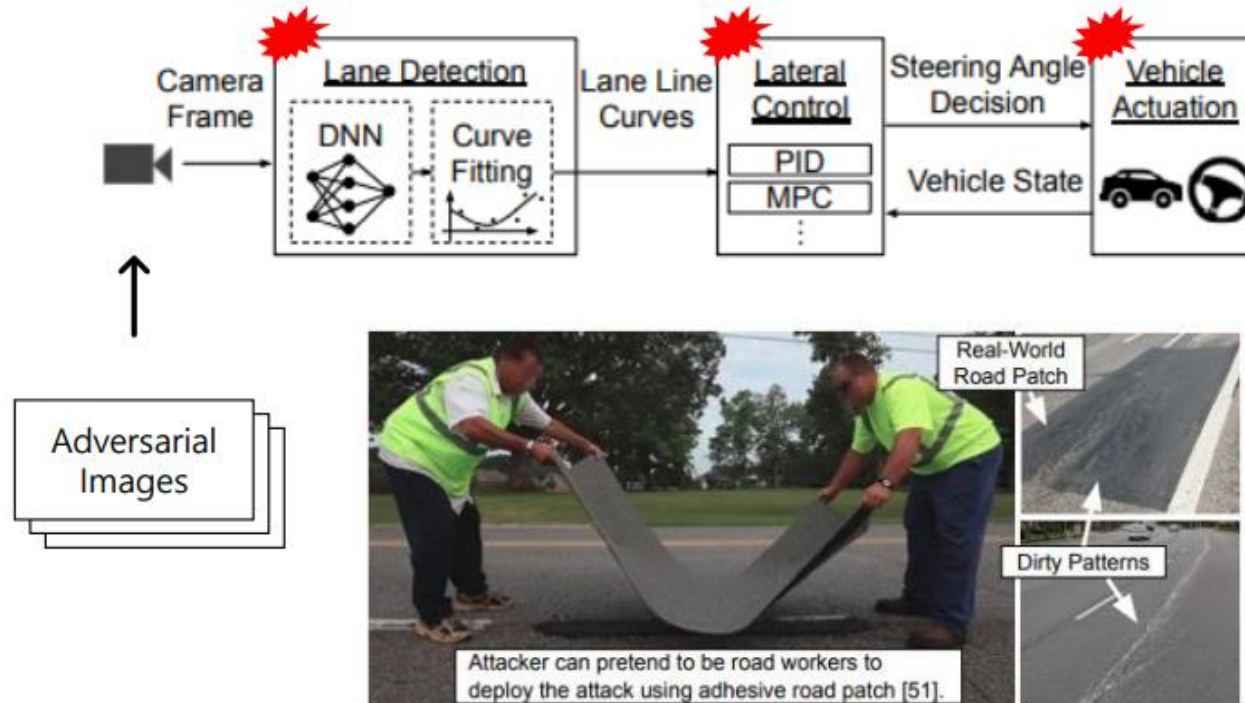
■ Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks

- Phantom 공격
 - Projector (with drone)를 통한 영상 송출
 - Digital screen을 통한 영상 송출
- Split-second Phantom 공격
 - ADAS 시스템에 인식될 수 있는 짧은 시간동안 이미지를 송출함
 - ADAS Camera에서 인식될 수 있는 최소의 시간동안 fake 이미지 생성함
 - ✓ 따라서, 인간의 눈으로는 이상현상을 감지하기 어려움
- Tesla ADAS 취약점
 - Color, Context, 그리고 Texture를 고려하지 않음
 - Tesla의 ADAS는 감지된 object의 depth를 확인하지 않음
 - ✓ Sensor Fusion: Radar 정보보다 Camera 이미지에 기반한 판단



■ Dirty Road Patch (DRP) Attack

- Openpilot의 LKAS (Lane Keeping Assist System) 시스템이 오작동 하도록 adversarial example을 생성하는 공격 방법론을 제안
- 여러 환경요인으로 인해 더러워진 도로의 패턴 (e.g., 흙, 얼룩 등)과 유사한 이미지를 생성하도록 최적화 문제를 정의하고, 실제 도로에 배치함으로써 공격 수행



03

자동차
사이버공격
대응기술

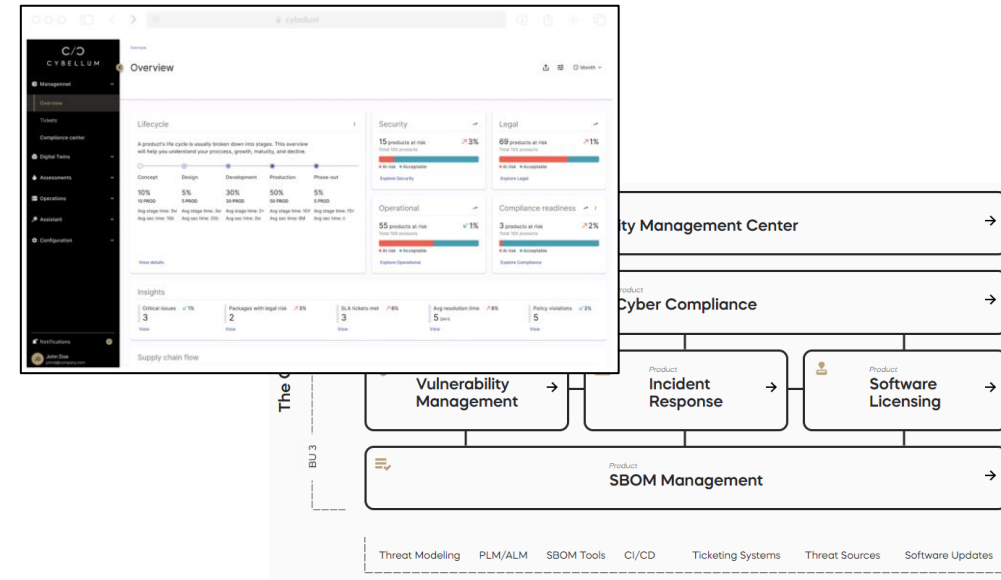
■ Analytics

• Automotive Vulnerability Analysis System (VAS)

- 해킹 방지를 위해 자동차 Firmware or Software에 대한 취약점 분석 기술

• CAN Bus Reversing

- CAN Data 기반으로 CAN ID 식별 등의 DBC 파일 정보 Reversing



출처: <https://cybellum.com/>

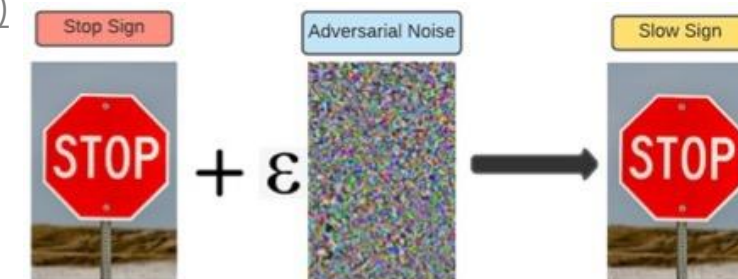
■ Detections

• Automotive Intrusion Detection System (IDS)

- In-vehicle or Vehicle to Back-end Infra에 대한 공격 침입 탐지 시스템
- 통신 Protocol에 의존적으로 연구되고 있으며, 최근에는 Automotive Ethernet 위주로 연구 진행

• Automotive AI Security (driver Identification, mitigation of AI adversarial attack)

- Driving Data를 기반으로 운전자 식별 관련 연구
- Poisoning attack 등 Automotive AI 모델의 Adversarial Attack에 대한 대응 기술 연구



출처: <https://insights2techinfo.com/self-driving-automobiles-and-adversarial-attacks/>

■ Cybellum

- 이스라엘의 자동차 사이버보안 전문 기업으로 LG전자가 미래성장동력으로 키우는 전장사업의 포트폴리오를 고도화하고 글로벌 경쟁력을 한층 더 강화하기 위해 2021년에 인수합병
- **자동차 내부 취약점 분석**
 - 사이버보안의 디지털 트윈(Digital Twin)
 - ✓ 소스코드 없이 바이너리 파일 내에서 자동차 소프트웨어의 모든 특성을 추출하여 SBOM*, 라이선스, 하드웨어 아키텍처, 운영 체제, 소프트웨어의 구성, 제어 흐름, API 호출 등을 포함한 각 차량의 구성요소를 분석
 - 인텔리전스(Intelligence) 기반 방어
 - ✓ 다양한 위협 인텔리전스 데이터베이스를 기반으로 새로운 취약점, 기존 위협에 대한 변경 사항 및 새로운 공격 방법을 추적
 - 위험기반 우선순위 지정
 - ✓ 차량의 각 구성 요소가 작동하는 취약점 간의 연관관계를 바탕으로 차량과 관련 없는 취약점을 걸러내고 취약점에 대한 우선 순위를 지정하여 개선 사항을 제공
- **국제 표준 준수 검증**
 - ISO/SAE 21434 준수 여부 검증 및 CERT C, MISRA C등 표준 준수 여부 검증

- SBOM(Software Bill Of Material)*: 최종 배포/활용되는 소프트웨어를 구성하는 모든 컴포넌트 명세서

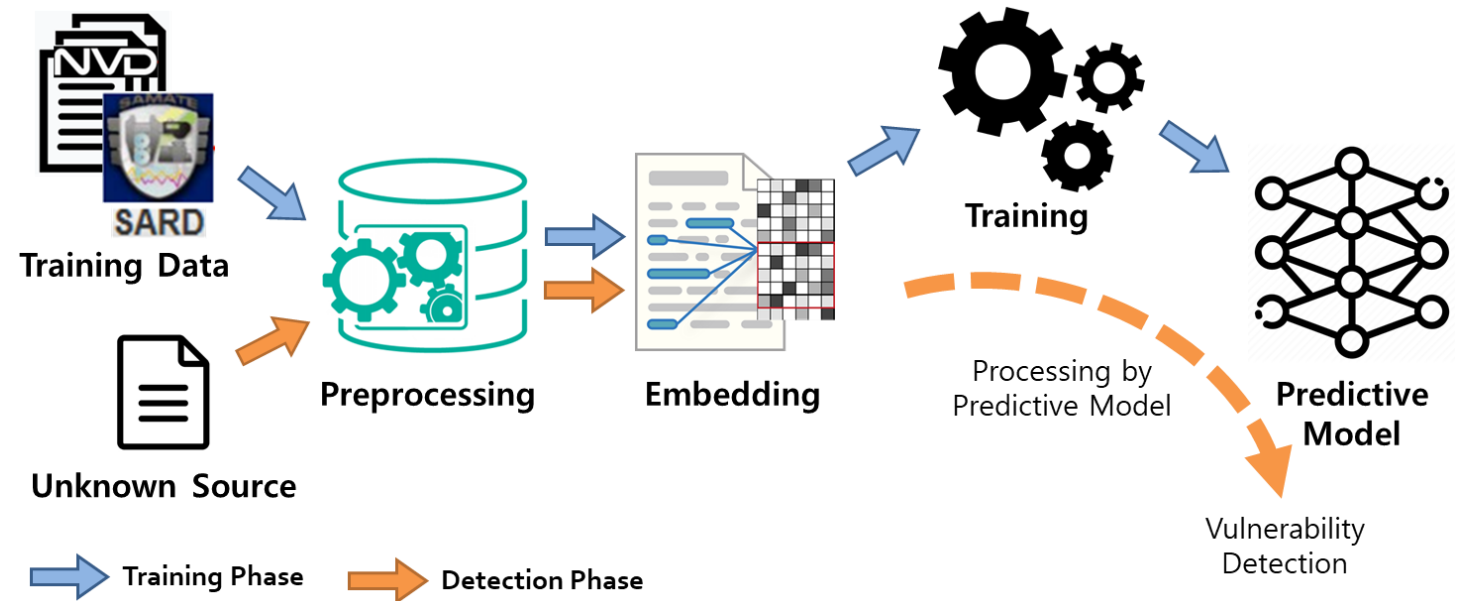
■ Automated Vulnerability Analysis System (AutoVAS)

- Dataset
 - NVD based on CVE**
 - SARD based on CWE***

- Source code Representation:
 - Program slicing
 - Symbolization & Tokenization
 - Source Code Embedding

• Neural Network

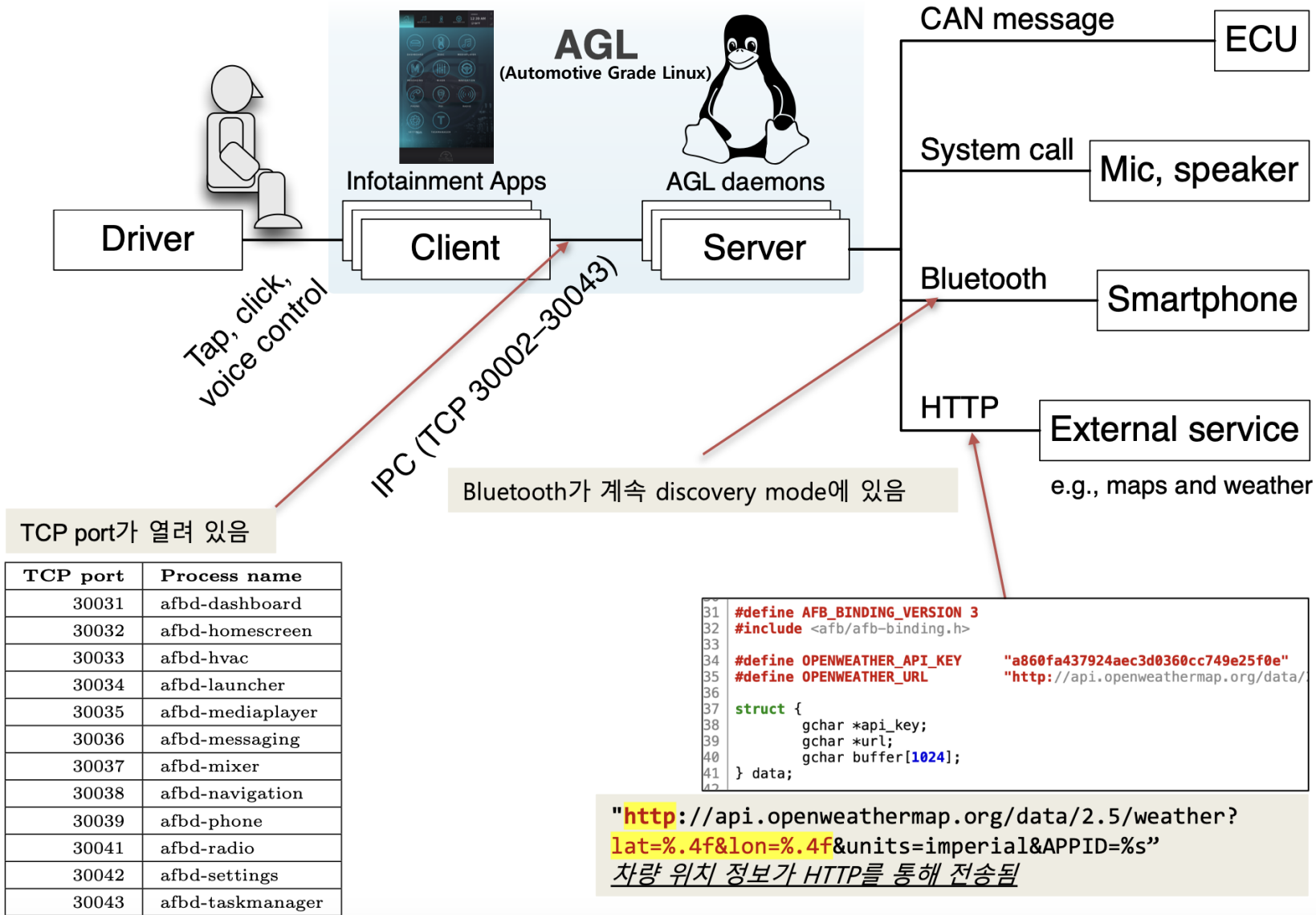
- RNN model
 - ✓ LSTM, GRU
- NLP model
 - ✓ BERT, GPT



< Workflow of AutoVAS >

- CVE**: Common Vulnerabilities and Exposure
- CWE***: Common Weakness Enumeration

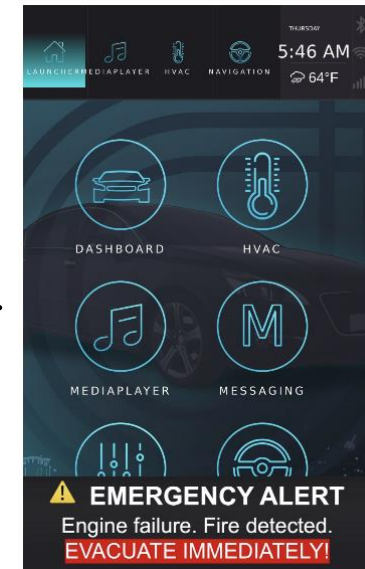
Findings



■ Attack#1. Command Injection Attack (CVE-2022-24595)

- Change temperature in HVAC service to trigger CAN messages

vcan0	7DF	[8]	02	01	0C	00	00	00	00	00	
vcan0	7DF	[8]	02	01	0D	00	00	00	00	00	
vcan0	7DF	[8]	02	01	0C	00	00	00	00	00	
vcan0	030	[8]	64	10	10	F0	0A	01	00	00	'd.....'
vcan0	7DF	[8]	02	01	0D	00	00	00	00	00	
vcan0	7DF	[8]	02	01	0C	00	00	00	00	00	
vcan0	7DF	[8]	02	01	0D	00	00	00	00	00	



- Change the current location to (0, 0) and send the location message manipulated by an external attacker

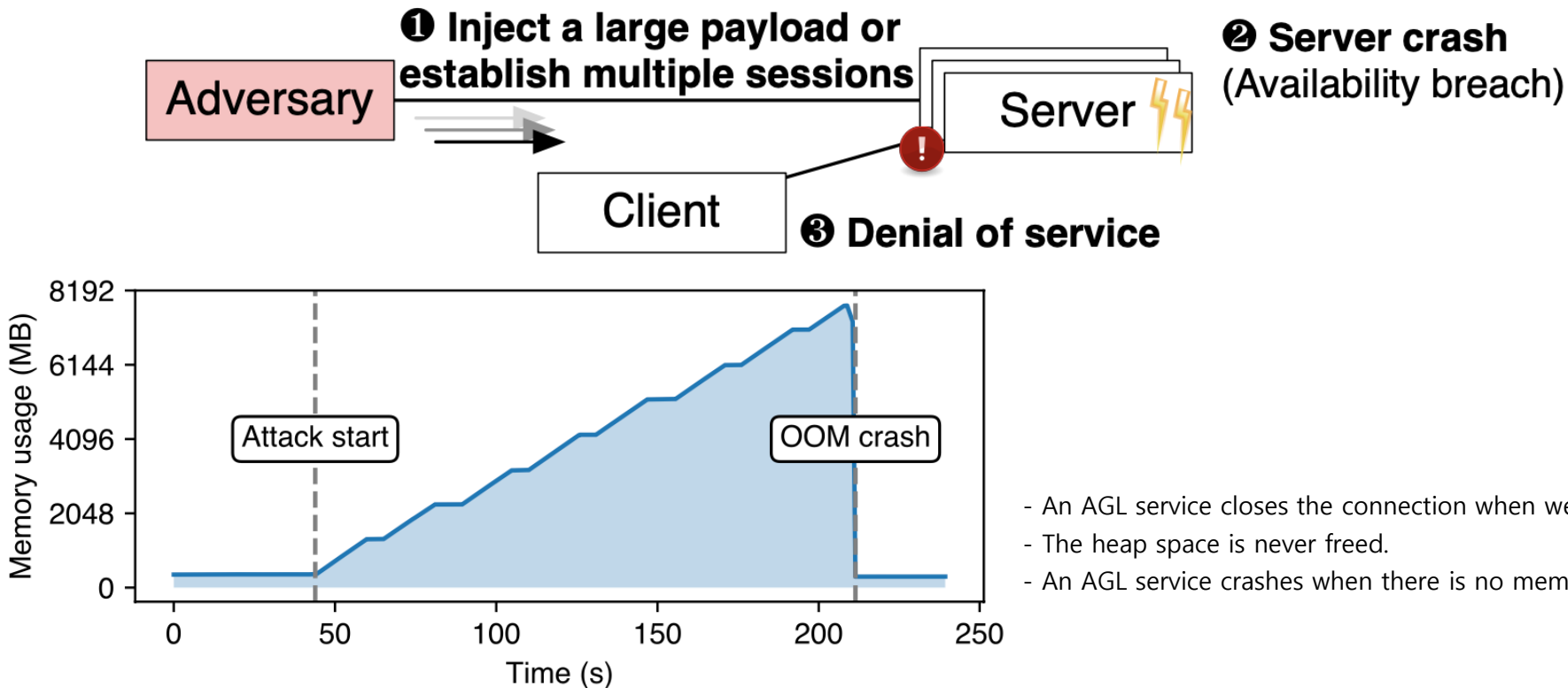
27	3.149379	127.0.0.1	127.0.0.1	TCP	98	42180 → 30038 [ACK] Seq=639 Ack=1183 Win=65280 Len=0 TSval=1420195774 TSecr=1420195774
28	51.060178	127.0.0.1	127.0.0.1	WebSoc...	272	WebSocket Text [FIN]
29	51.060277	127.0.0.1	127.0.0.1	TCP	98	42180 → 30038 [ACK] Seq=639 Ack=1183 Win=65280 Len=0 TSval=1420195774 TSecr=1420195774
30	51.065616	127.0.0.1	127.0.0.1	WebSoc...	259	WebSocket Text [FIN]
31	51.065701	127.0.0.1	127.0.0.1	TCP	98	42180 → 30038 [ACK] Seq=639 Ack=1183 Win=65280 Len=0 TSval=1420195774 TSecr=1420195774
32	68.916154	127.0.0.1	127.0.0.1	WebSoc...	285	WebSocket Text [FIN]
33	68.916918	127.0.0.1	127.0.0.1	WebSoc...	193	WebSocket Text [FIN]
34	68.917020	127.0.0.1	127.0.0.1	TCP	98	42180 → 30038 [ACK] Seq=639 Ack=1183 Win=65280 Len=0 TSval=1420195774 TSecr=1420195774
35	68.917355	127.0.0.1	127.0.0.1	WebSoc...	272	WebSocket Text [FIN]
36	68.917427	127.0.0.1	127.0.0.1	TCP	98	42180 → 30038 [ACK] Seq=639 Ack=1183 Win=65280 Len=0 TSval=1420195774 TSecr=1420195774
37	68.917600	127.0.0.1	127.0.0.1	WebSoc...	191	WebSocket Text [FIN]
38	68.917662	127.0.0.1	127.0.0.1	TCP	98	42180 → 30038 [ACK] Seq=639 Ack=1183 Win=65280 Len=0 TSval=1420195774 TSecr=1420195774
39	68.917889	127.0.0.1	127.0.0.1	WebSoc...	272	WebSocket Text [FIN]
40	68.917949	127.0.0.1	127.0.0.1	TCP	98	42180 → 30038 [ACK] Seq=639 Ack=1183 Win=65280 Len=0 TSval=1420195774 TSecr=1420195774
41	68.925671	127.0.0.1	127.0.0.1	WebSoc...	272	WebSocket Text [FIN]
42	68.926384	127.0.0.1	127.0.0.1	WebSoc...	272	WebSocket Text [FIN]
43	68.926475	127.0.0.1	127.0.0.1	TCP	98	42180 → 30038 [ACK] Seq=639 Ack=1183 Win=65280 Len=0 TSval=1420195774 TSecr=1420195774
44	68.926707	127.0.0.1	127.0.0.1	WebSoc...	272	WebSocket Text [FIN]
45	68.926777	127.0.0.1	127.0.0.1	TCP	98	42180 → 30038 [ACK] Seq=639 Ack=1183 Win=65280 Len=0 TSval=1420195774 TSecr=1420195774

Wireshark - Packet 28 - attack.pcap	
> Frame 28: 272 bytes on wire (2176 bits), 272 bytes captured (2176 bits)	
> Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)	
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1	
> Transmission Control Protocol, Src Port: 30038, Dst Port: 42180, Seq: 1183, Ack: 639, Len: 174	
> WebSocket	
Line-based text data (1 lines)	[5,"navigation/position",{"event":"navigation","latitude":36.1363,"longitude":-115.151,"position":"car"},"afb-event"]

Wireshark - Packet 30 - attack.pcap	
> Frame 30: 259 bytes on wire (2072 bits), 259 bytes captured (2072 bits)	
> Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)	
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1	
> Transmission Control Protocol, Src Port: 30038, Dst Port: 42180, Seq: 1357, Ack: 639, Len: 161	
> WebSocket	
Line-based text data (1 lines)	[5,"navigation/position",{"event":"navigation","latitude":0,"longitude":0,"position":"car"},"afb-event"]

■ Attack#2. DoS Attack (CVE-2022-24596)

- CVE-2022-24596 due to Improper Input Validation & Resource Management
 - AGL services could be crashed with an out-of-memory (OOM) crash.
 - ✓ After the OOM crash, a driver cannot utilize infotainment services until reboot.



■ Attack#3. MITM* attack (CVE-2022-24597)

- CVE-2022-24597 due to Insecure communication over HTTP
- AGL tries to connect OpenStreetMap.org via HTTP communicatinos

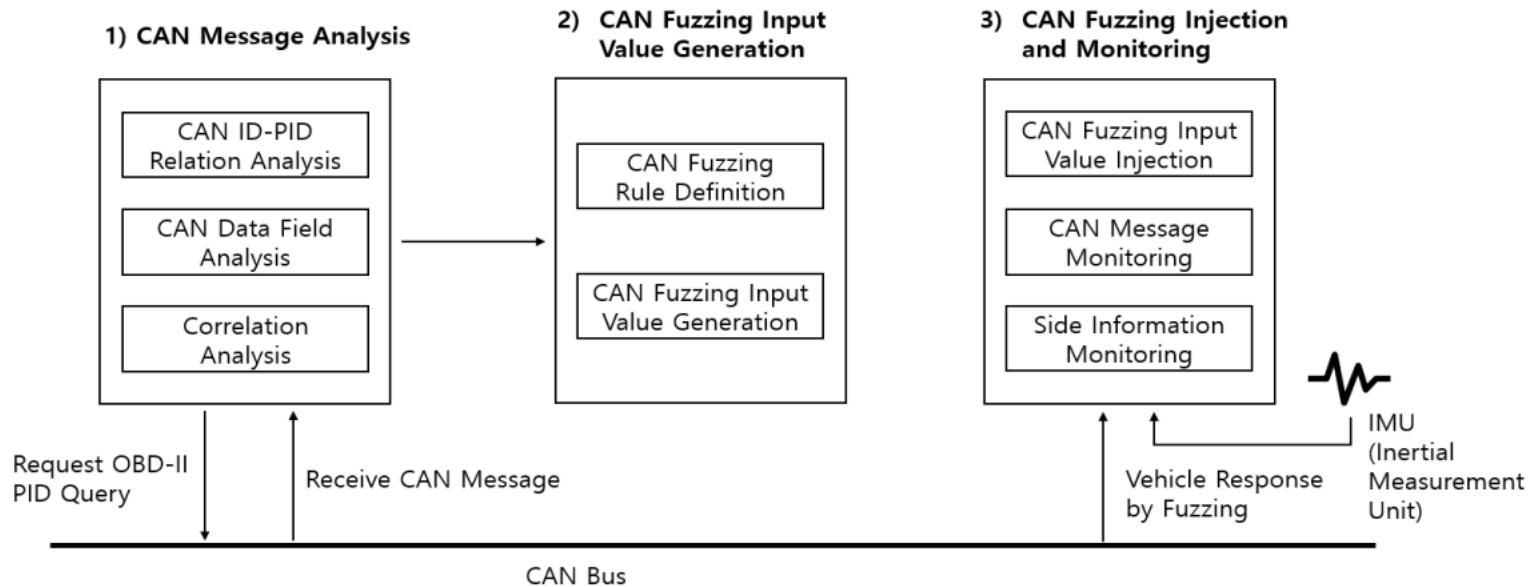


- Since there is no client authentication, an unauthorized client can hijack various vehicle information.

- MITM*: Man In The Middle

■ Efficient ECU Analysis Technology Through Structure-Aware CAN Fuzzing

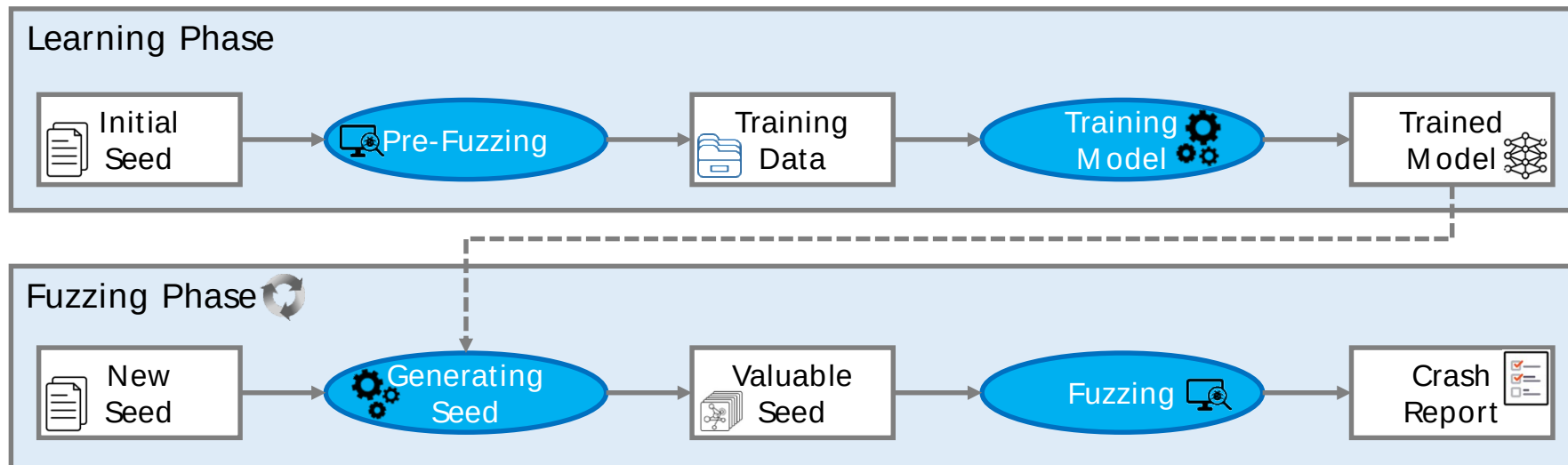
- CAN 메시지 특성이 반영된 Rule 기반으로 CAN Fuzzing 입력 값을 생성하고, Fuzzing 결과를 모니터링할 수 있는 방법론을 최초로 제시
 - CAN 메시지의 특성이 반영된 Non-Random CAN Fuzzing 기술을 제안함으로써, 기존 Random CAN Fuzzing 기술들에 비해 CAN Fuzzing에 소요되는 시간을 절약할 수 있음



[Ref] 김동영, 전상훈, 류민수 and 김휘강. (2022). 자연어 처리 모델을 활용한 퍼징 시드 생성 기법. 정보보호학회논문지, 32(2), 417-437.

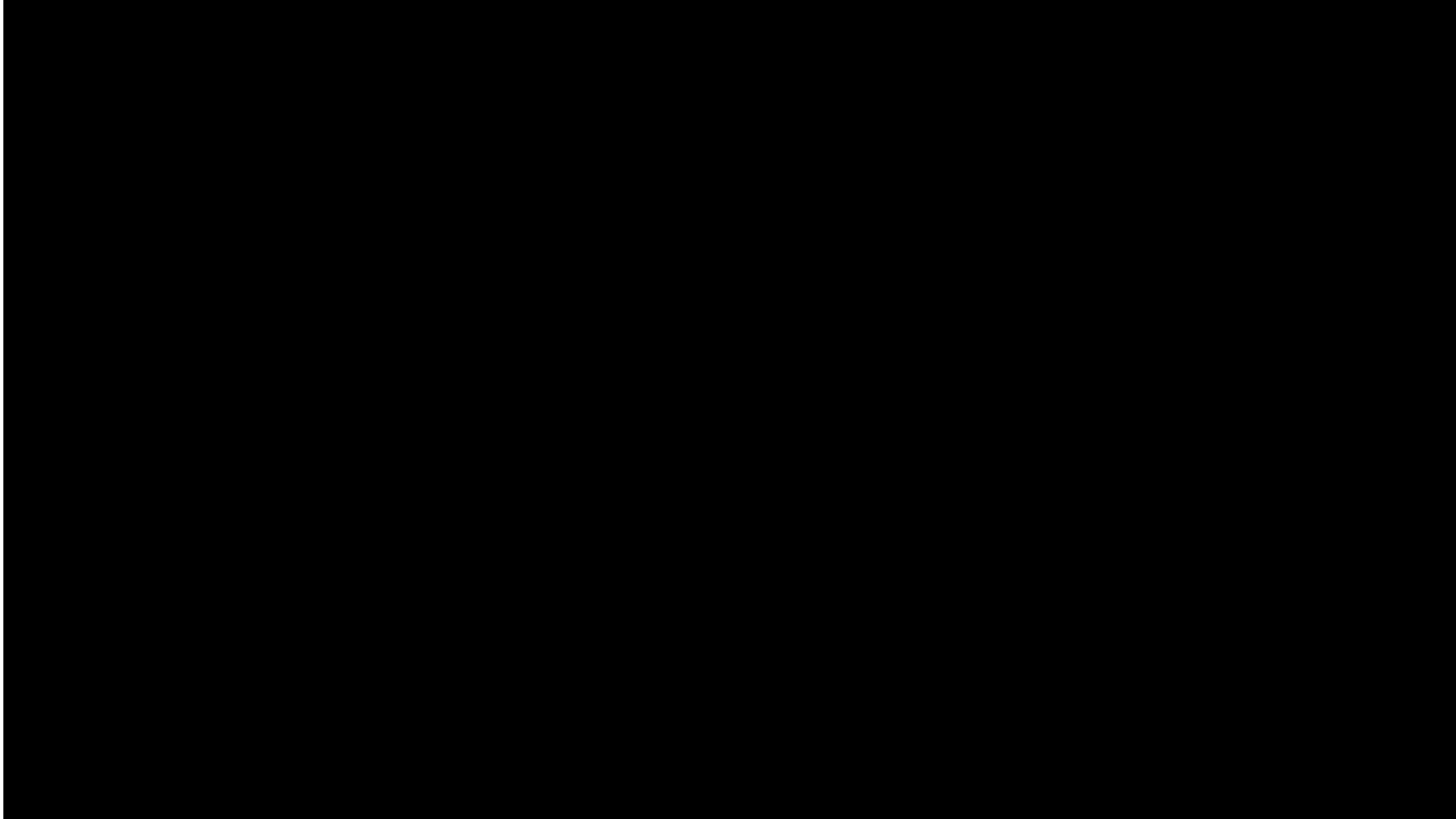
■ Data-DRiven Fuzzing (DDRFuzz)

- Learning Phase
 - To gather training data using pre-fuzzing (triggering crash and new coverage)
 - A seed generative model based on a sequence-to-sequence (seq2seq) model
- Fuzzing Phase
 - To obtain valuable seeds through trained seq2seq model
 - Easy to integrate with several base fuzzers since separation between learning and fuzzing phase.



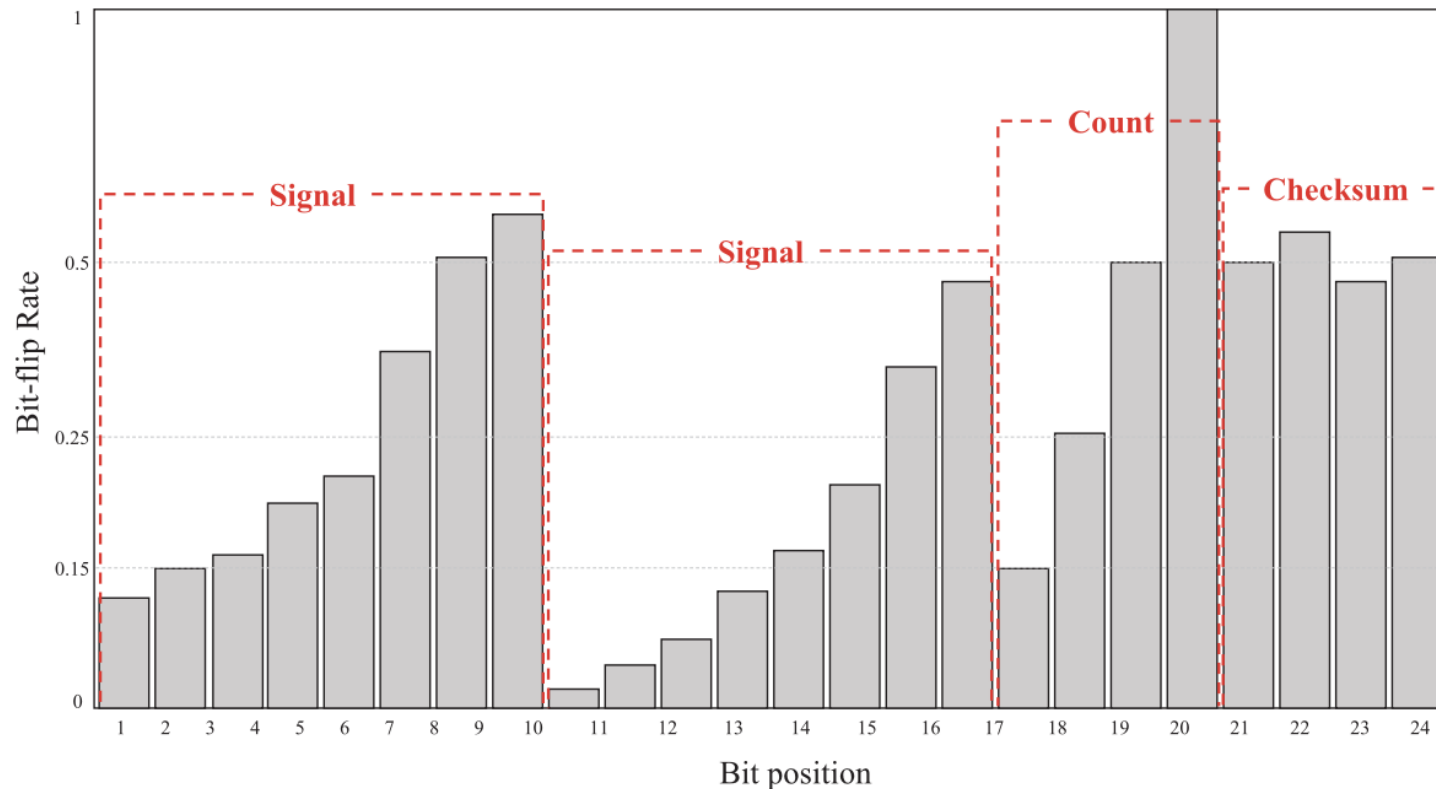
< Workflow of DDRFuzz >

■ Fuzzing Demo

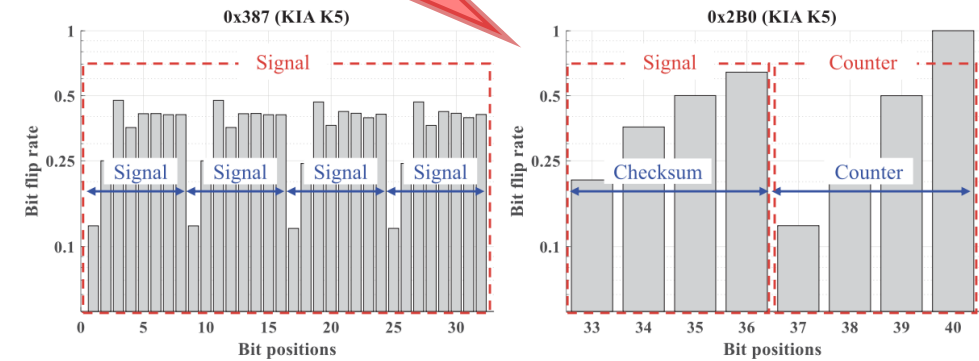


■ CAN Data 식별#01

- CAN Data의 Bit Flip Rate를 기준으로 경계 값을 식별
 - Bit Flip Rate가 순차적으로 증가하다가 경계 값을 지나면 감소하는 pattern을 기준으로 CAN Data를 식별하는 방법



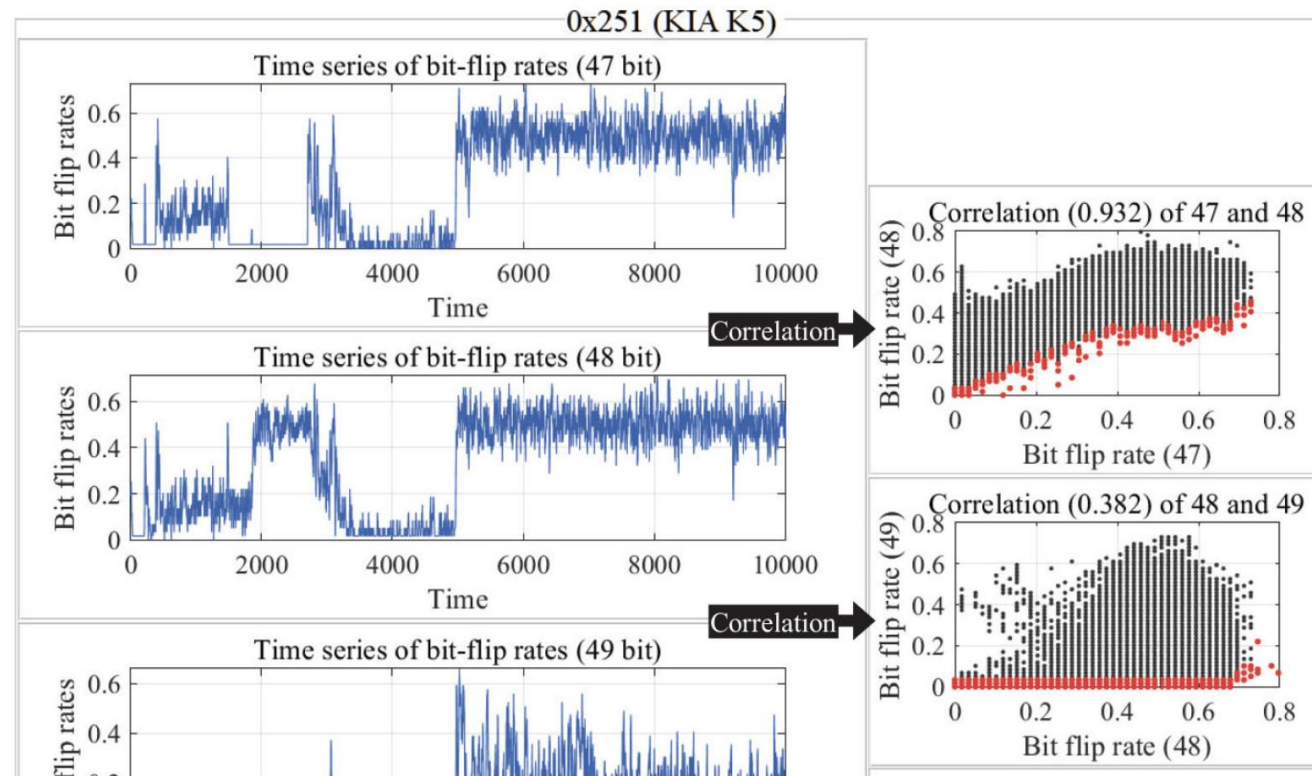
Counter Examples



[Ref] Choi, Wonsuk, et al. "An enhanced method for reverse engineering CAN data payload." IEEE Transactions on Vehicular Technology 70.4 (2021): 3371-3381

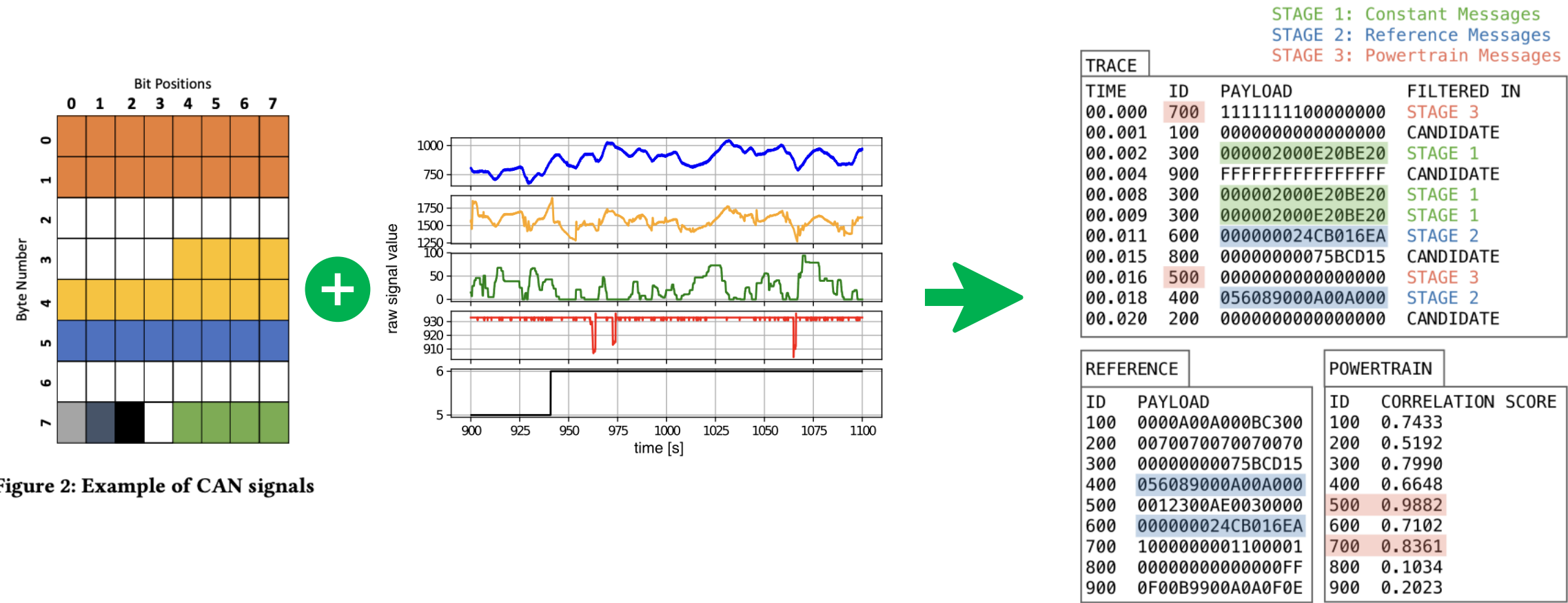
■ CAN Data 식별#02

- CAN Data의 Bit flip rate 대신 time series의 pattern을 이용해서 경계 값 식별
 - Bit position 대신 time을 기준으로 x축을 구성 후에 time-series의 pattern을 기준으로 CAN Data를 식별하는 방법
 - Bit flip의 counter example의 경우에도 time series로 변환 후 경계값 식별됨을 확인



■ LibreCAN

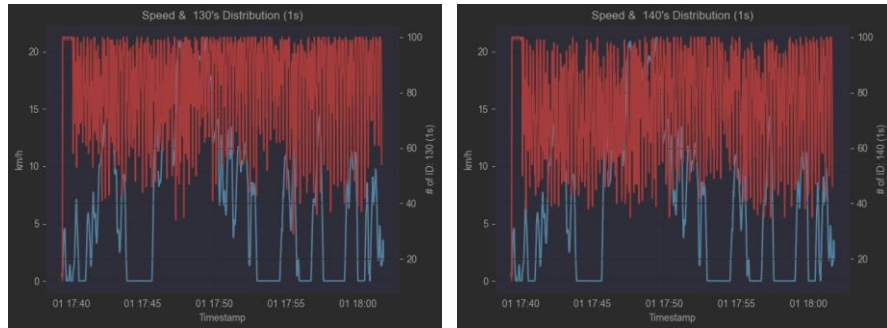
- 앞선 CAN Data 경계 값 식별 방식을 기반으로 DBC2-PID 표준을 결합하여 CAN Message 구조 유추



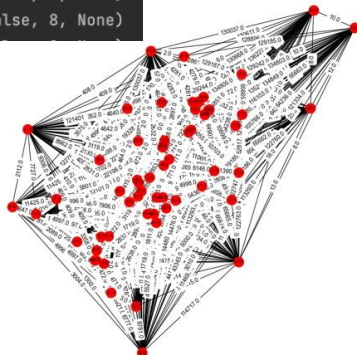
[Ref] Pesé, Mert D., et al. "LibreCAN: Automated CAN message translator." Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019.

■ DTW(Dynamic Time Wrapping)와 GCN(Graphic Convolution Network)을 이용하여 CAN ID 식별

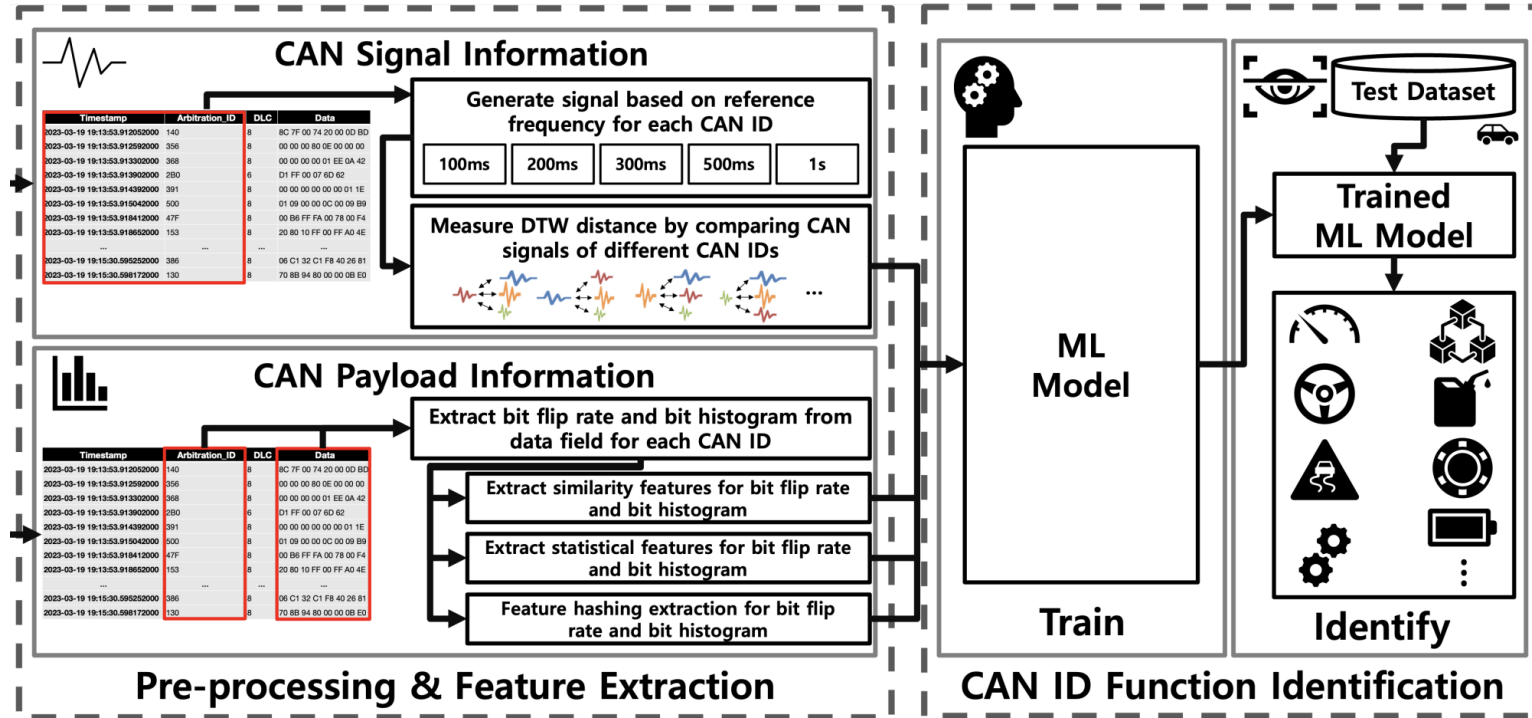
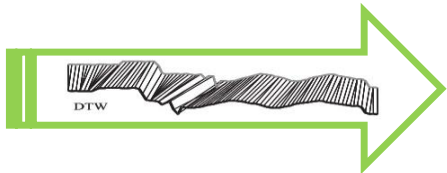
- 기존 방법은 시계열 데이터의 align이 맞지 않는 경우 적용하기 어렵다는 단점을 보완하기 위해 연구
- DTW를 사용해서 여러 시계열 데이터의 유사도를 동적으로 분석



```
message('YRS11', 0x130, False, 8, None)
message('YRS12', 0x140, False, 8, None)
```

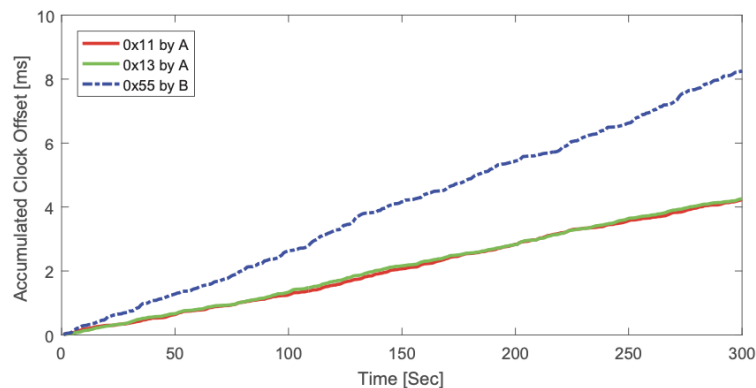


<< GCN Model >>

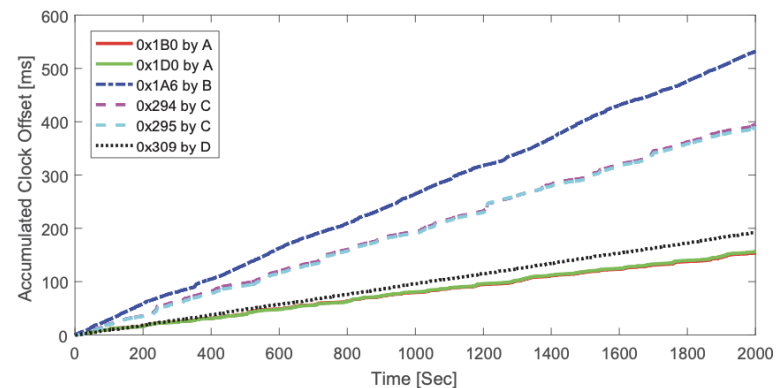


■ Clock-based IDS (Intrusion Detection System)

- 자동차 내부 네트워크에서 ECU가 CAN 통신 프로토콜을 이용하여 주기적으로 메시지를 전송할 때, ECU의 Clock 생성기의 하드웨어 특성이 미세하게 다르다는 점을 이용
 - 자동차 내부 네트워크의 여러 ECU들은 항상 자신들의 Unique한 clock으로 time을 counting하기 때문에, clock offset을 계속 누적하여 선형 그래프로 표현되고, 각 ECU의 선형 그래프는 고유의 기울기를 가짐
 - 이러한 선형 그래프의 기울기를 Clock Skew라고 하고, Clock Skew를 계산함으로써 ECU Fingerprinting이 가능해 짐
- 즉, Clock Skew 기반의 ECU Fingerprinting 기술을 이용하여 CAN Traffic의 이상 유무를 판단



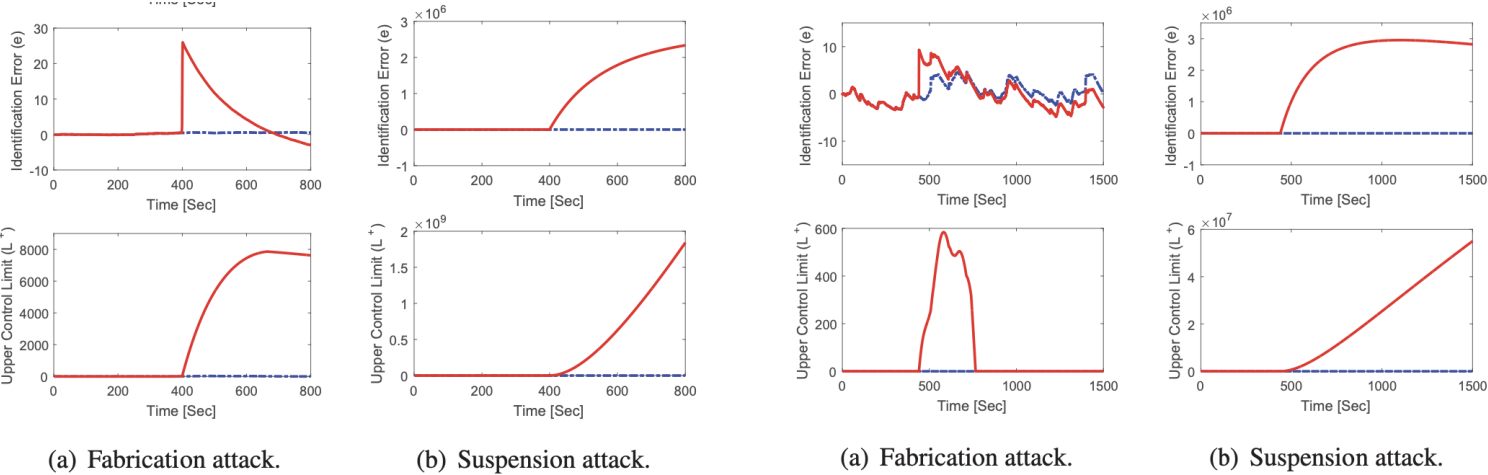
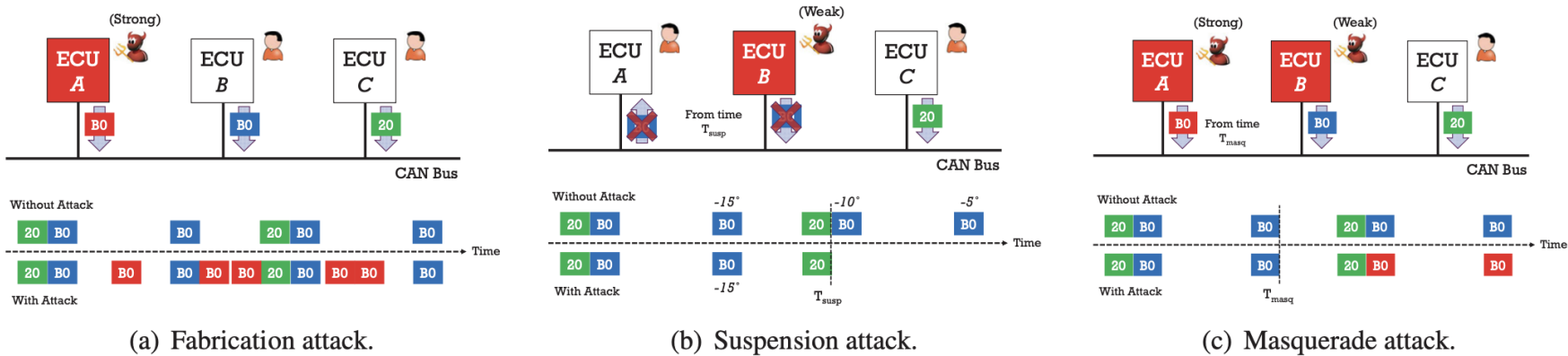
(a) CAN bus prototype.



(b) Honda Accord 2013.

■ Clock-based IDS (cont'd)

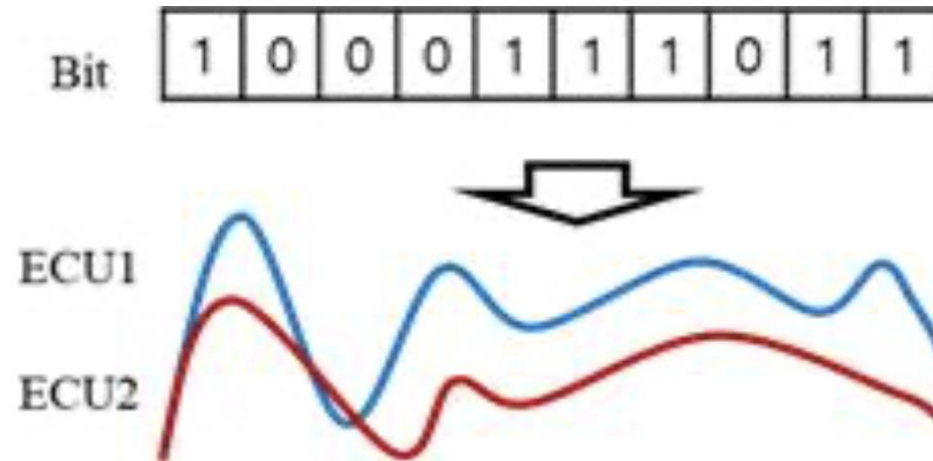
- Representative Attack for in-vehicle



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

■ Voltage-based IDS

- ECU의 전력신호를 분석하여 Fingerprinting하는 기술을 Automotive IDS에 적용
 - CAN 통신 프로토콜은 CAN-H와 CAN-L라고 불리는 전기선에 전압차를 줌으로써 비트를 표현
 - 2가닥 전기선에 약 2.5v 전압차가 있다면 비트 0을 표현하고, 전압차가 없는 idle 상태라면 비트 1을 표현
 - 이처럼 비트 시퀀스를 아날로그 신호로 인코딩하는 signaling 과정에서 각 ECU의 하드웨어 특성으로 인해 아날로그 신호가 미세하게 차이 발생하며, 이러한 특징을 이용해 ECU Fingerprinting에 활용



■ DATA 필드를 포함한 이상여부 탐지

- 대상 차량에 대해 보다 정확한 탐지 정확도
- 공격 탐지 결과에 대한 자세한 정보를 얻을 수 있음
 - 어떤 공격이 발생했는지? (DoS, Fuzzy, ...)
 - 무엇을 위한 공격인지? (차량 문 잠금,...)

■ DATA 필드를 제외한 이상여부 탐지

- 보다 빠른 탐지 속도
- 여러 차량에 대해 적용 가능한 범용적 방법론
- 종류
 - Time interval-based IDS
 - Entropy-based IDS

■ Time-interval based IDS

- 메시지 발생 주기를 이용해 공격 탐지
 - 차량 시스템에서 발생하는 Arbitration ID별 평균 메시지 발생 횟수 측정
 - 각 ID별 정상 수신 interval을 정의
 - 실제 메시지를 수신하면서, 임계 값 초과 시 마다 Score 조정
 - Score가 과도하게 높아지면 주입 공격에 대한 경고를 알림

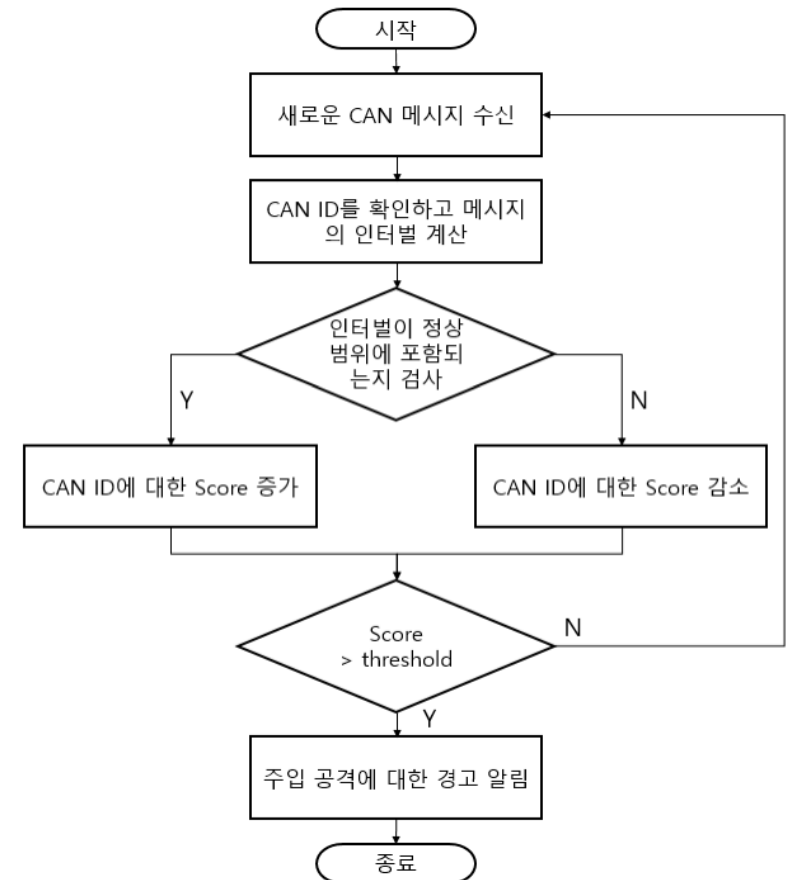
Normal Status: Immediate response (within reasonable time)

```
Timestamp: 0.651241 ID: 0153 100 DLC: 0
Timestamp: 0.651485 ID: 0153 000 DLC: 8 00 80 10 ff 00 ff 40 ce
```

Under Attack Status: Remote Frame 01F1 is requested but get no response (within reasonable time)

```
Timestamp: 0.254877 ID: 01f1 100 DLC: 0
Timestamp: 0.255148 ID: 0080 000 DLC: 8 00 17 c8 09 19 11 19 8d
Timestamp: 0.255230 ID: 0000 000 DLC: 0
Timestamp: 0.255479 ID: 0081 000 DLC: 8 7f 84 8e 00 00 00 00 bd
Timestamp: 0.255576 ID: 0000 000 DLC: 0
Timestamp: 0.255824 ID: 018f 000 DLC: 8 00 36 19 00 00 3f 00 00
Timestamp: 0.255922 ID: 0000 000 DLC: 0
```

(In general, offset is within 5, Time Interval is 0.001196 by car vendor's default setting.)



■ Entropy-based IDS

- 대상 차량 내 모든 메시지 ID에 대한 엔트로피(H')를 계산
- DoS 공격 시 내부 네트워크의 엔트로피가 감소함에 따라 공격을 탐지
- DoS 공격 시 특정 ID에 대한 점유율이 올라가기 때문에 불확실성이 낮아짐을 확인
- 특정 ID외 CAN Bus에서 사용되지 않은 ID들을 이용하여 임의의 메시지들을 삽입할 경우 불확실성은 증가하여 엔트로피 값은 높아지는 것을 확인
- 따라서 엔트로피 값의 정상 범위를 의미하는 상한선과 하한선을 지정하고 시간 단위 별 엔트로피 값을 모니터링하는 이상 탐지 시스템을 설계

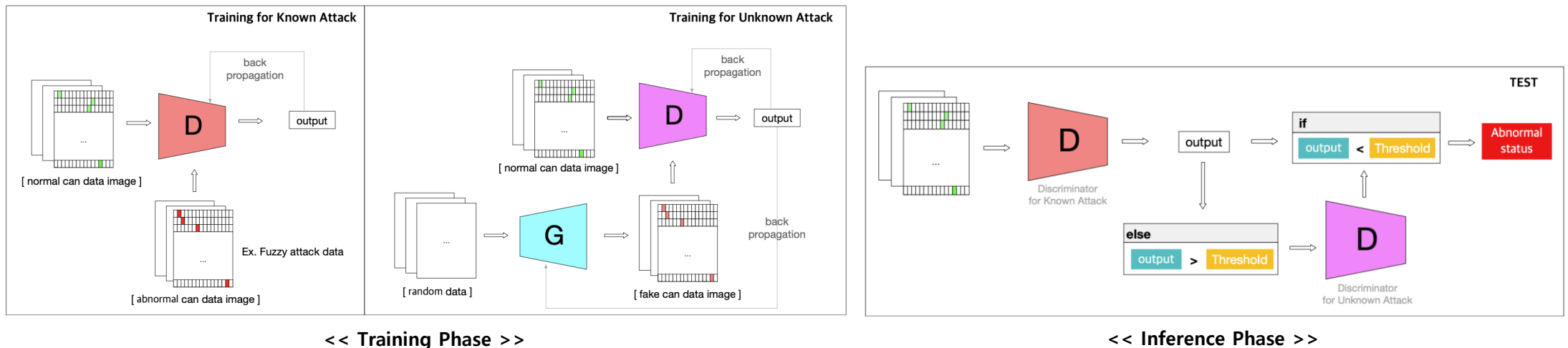
$$H' = - \sum_{i=1}^n p_i \log p_i$$

n : 차량 내부 네트워크에서 사용되는 CAN ID 수
 p_i : 전체 메시지 중 특정 CAN ID의 비율($0 \leq p_i \leq 1$)

CAN ID	사용 횟수	p_i	$\log p_i$	H^i
0x153	2	0.003 ↓	-5.81 ↑	점차 낮아짐
0x164	8	0.012 ↓	-4.42 ↑	
0x2c0	6	0.009 ↓	-4.71 ↑	
...				
0x000	284	0.850 ↑	-0.160 ↓	

■ AI based IDS

- GAN based IDS
 - To respond unknown attacks, GAN can be one of the best solution to detect undiscovered/unknown attacks
 - Known attack detection
 - ✓ Training the 1st discriminator using known attack and normal data
 - Unknown attack detection
 - ✓ Training the 2nd discriminator using only normal data without attack data
 - ✓ The 2nd discriminator is trained to distinguish between real data and fake data



<< Training Phase >>

<< Inference Phase >>

■ AI based IDS

• Automotive Ethernet에 대한 요구사항

- **잡음:** 통신 회선에서 잡음을 적게 방출시켜야 하며, 외부로부터의 잡음에 robust 해야함
- **우선 순위:** 전달되는 payload의 우선순위를 매길 수 있어야 하며, 해당 payload는 반드시 정해진 시간 내에 처리되어야 함
 - ✓ Powertrain 계열의 데이터는 다른 데이터보다 우선 처리하여 반드시 예상되는 지연시간 내에 도착해야 함
 - ✓ Powertrain 관련 데이터를 기존처럼 Best-effort로 처리하면, 자동차의 특성상 사고 발생 가능성 존재
- **대역폭 예약:** 특정 서비스를 위해 사용 가능한 대역폭의 일부를 항상 사용 가능 하도록 유지해야 함
 - ✓ ADAS와 연동된 카메라는 항상 일정한 스트림을 유지하여 처리 장치에 영상 정보를 전달해야 함
 - ✓ 오디오 데이터는 항상 일정한 대역폭을 사용해 스피커에 데이터를 전달해야 함
- **동기화:** 동일 네트워크로 연결된 장치들은 매우 적은 오차로 시간을 맞추 수 있어야 함
 - ✓ ADAS와 연동된 여러 센서들의 데이터는 정확한 시간으로 동기화 되어야 올바른 처리가 가능함
 - ✓ 차량에 탑재된 10개 이상의 스피커는 동시에 같은 소리를 재생할 수 있어야 함

- I. SDV 개념의 확산으로 인해 E/E Architecture가 점차 중앙 집중형으로 변화됨에 따라 보안 위협은 더욱 커질 것
- II. 자동차를 대상으로 하는 공격은 주로 네트워크와 센서를 이용하는 공격이 많음. 최근에는 자율주행 자동차의 센서 공격에 대한 연구가 많아지고 있음.
- III. 대응 방안에 대한 연구는 크게 두 가지로 구분되어 진행됨
 - ✓ 자동차 취약점 분석 및 위협 시나리오 개발 (Automotive VAS and Exploit Scenarios)
 - ✓ 위협 탐지 시스템 (Intrusion Detection System)

Thank You



- Lab: <https://mose.kookmin.ac.kr>
- Email: sh.jeon@kookmin.ac.kr