


Automotive scary signals and possible RCEs

By Sébastien Dudek

DEFCON Paris – November 2023



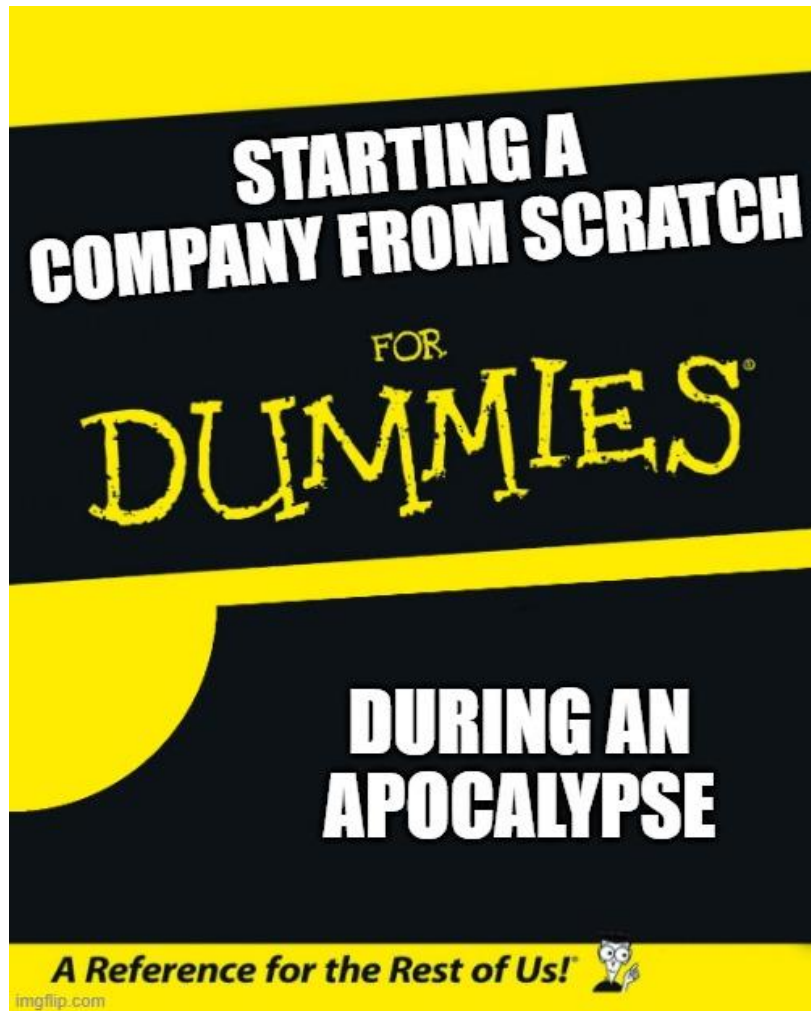
Founder of Penthertz

- Sébastien Dudek ([@FIUxluS](#))
 - CTO of Penthertz (as Chief Taxes Officer...)
 - Specialized in Wireless communications security
 - > 10 years of experience in Software & Hardware security
 - Security researcher
 - Pentester & Red Team
 - Vulnerability researcher
- Perfect mix to make Penthertz!**
- 

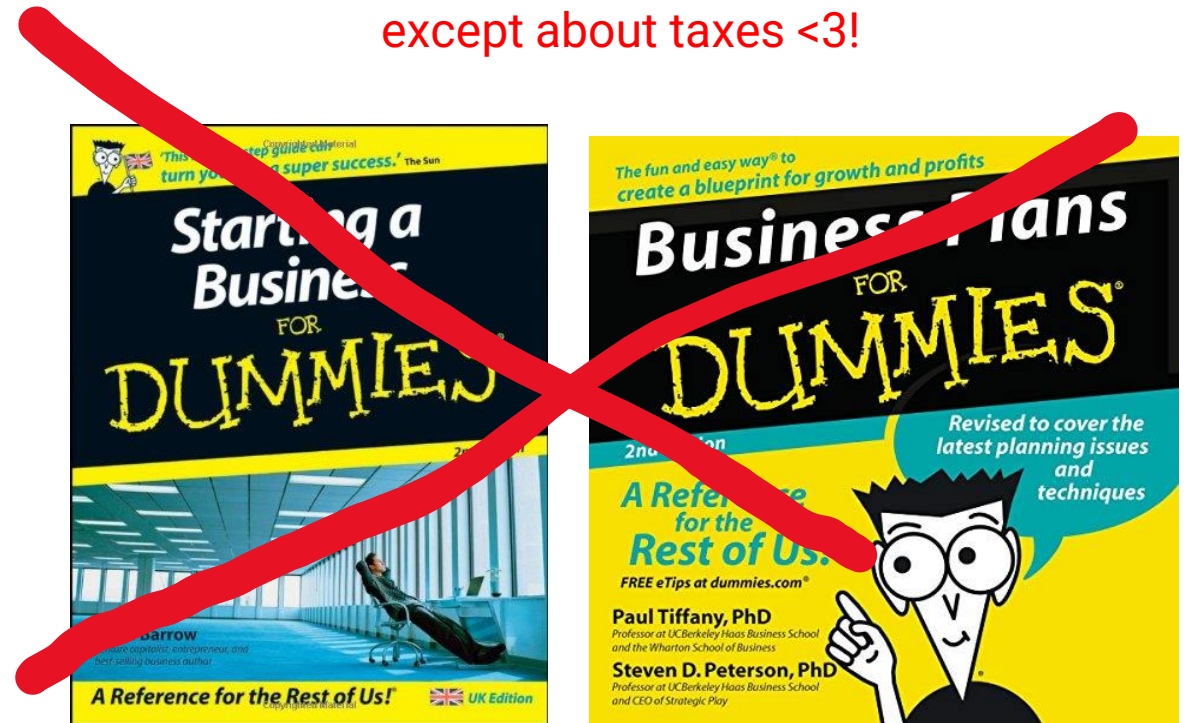


Future book idea (I'm still trying to sell...)

My next book (or not)



Forget everything you learned before,
except about taxes <3!



*but people have seen worse in restaurants... 😊

Main activities



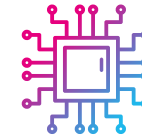
Security assessments

- Wireless communications (RFID, Wi-Fi, Mobile communications, Bluetooth, etc.)
- Embedded devices
- Backend servers
- Red Team



Trainings

- Software-Defined Radio Hacking
- Wi-Fi Red teaming
- RFID Hacking
- Mobile attacks (2G/3G/4G/5G), and more...



Hardware security

- Firmware extraction
- Chip off
- Secrets extraction
- Library's analysis
- Vulnerability hunting

Setup to PWN the radio



Part of the SDR material

- Need to manage any type of transmission (2G-5G, Wi-Fi, Remotes, Bluetooth, ZigBee, RFID, **exotic communications**, etc.).
- Today's challenges: handling from DC to 6 or even 8 GHz with a decent stability
- Next challenges → 30 GHz at least with mmWave bands
- Able to get large bandwidth in some situation (sometimes > 100 Msps even ≥ 300 Msps)



2021 Picture → the tables have never been so clean!

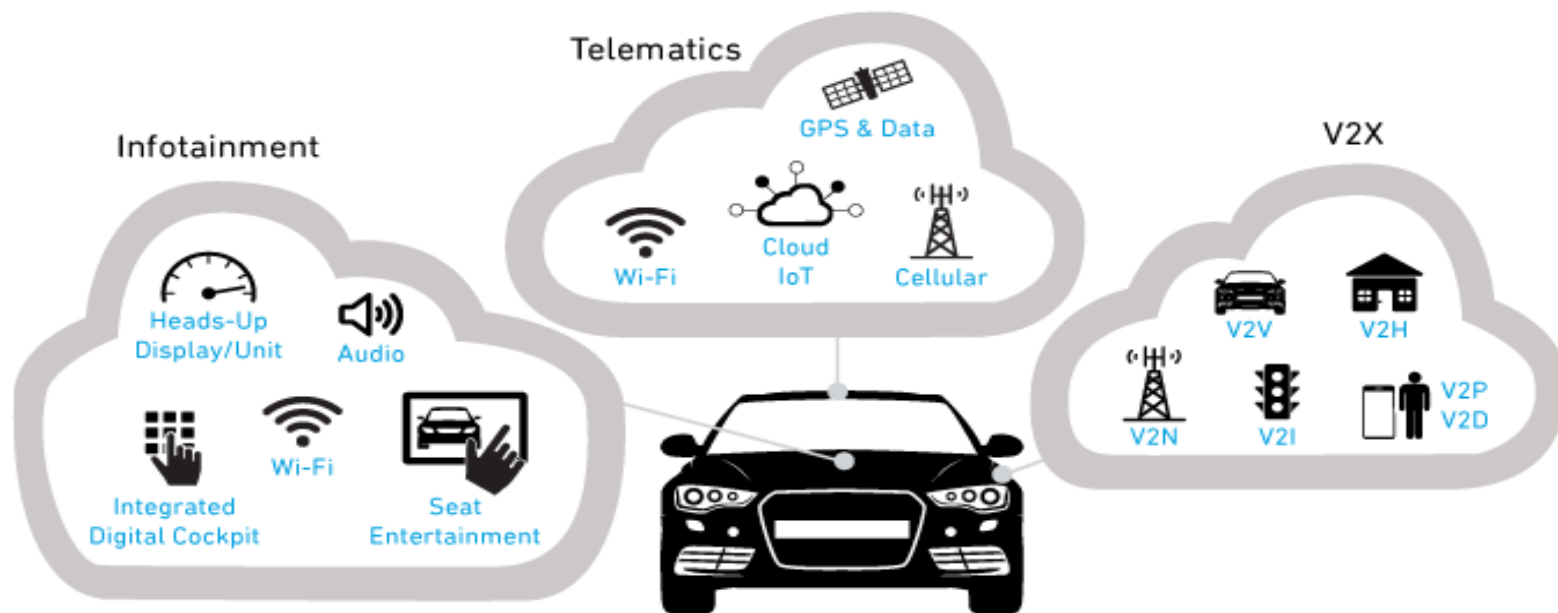
SDR has also performance limits to overcome, but let's talk about 5G use case in IoT!



Connected cars

RF communications in cars

Heterogeneous Connectivity



qorvo.

©2018 Qorvo, Inc.



Summary

- 01** Common vectors & attacks
- 02** Nice opportunities to debug OTA and get root
- 04** Going further
- 05** Conclusion



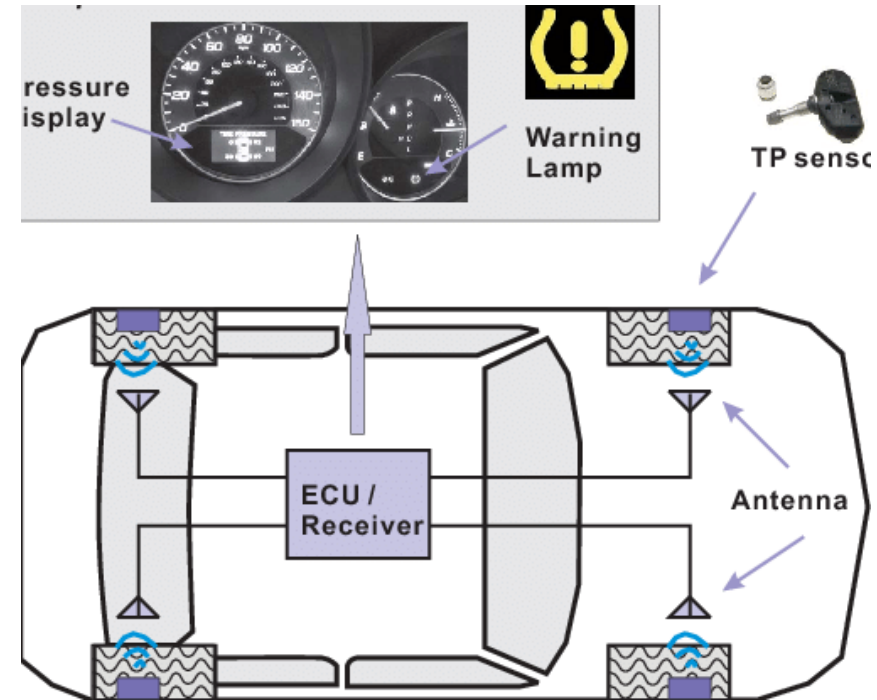
Vectors & Attacks

The background features a gradient from magenta on the left to blue on the right, overlaid with two sets of concentric circles. The left set is magenta and the right set is blue. The text 'TPMS' is centered in white.

TPMS

Introduction

- TPMS (Tire Pressure Monitoring System)
- 2 types/technologies:
 - Indirect → measurement of each wheel rate revolution
 - Direct → actual pressure level measurement



TPMS architecture with four antennas
(source: [1])

[1] Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Stud by Ishtiaq Rouf et al.

RF Signal

- Frequency bands -> ISM bands of the country mostly:
 - 433 MHz / 868 MHz in EU
 - 315 MHz / 433 MHz in US
 - Etc.
- Modulations:
 - ASK: Amplitude Shift Key
 - or 2-FSK/BFSK
 - or both (hybrid)



TPMS reader/trigger

Capture

- Using a Software-Defined Radio (SDR) device*
- SDR → more flexibility
- Supporting the range of frequency + adapted antenna
- Cheap option: RTL-SDR v3 for 30-50€ (but only for RX)



*Dedicated RF chips can be used instead of SDR = cheaper (~10€)

Decoding the data

Enter pattern here Search

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	1	d	4	8	5	3	d	7	4	5	e	3	2	f	f	f	f	6	5

selected

ol

ol

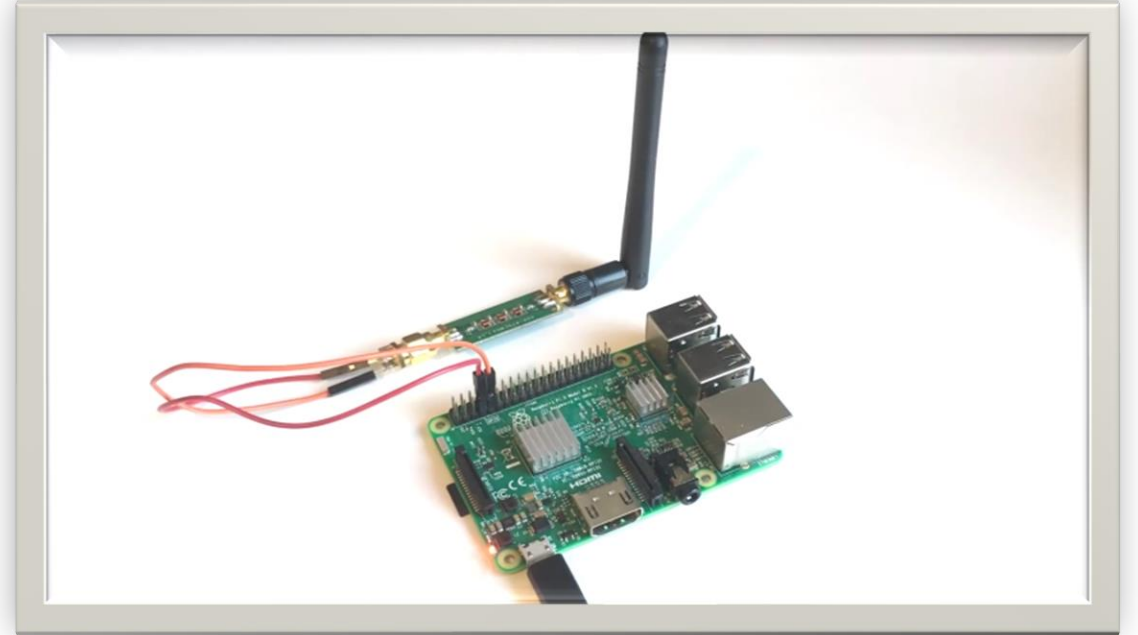
Bit: Hex: Decimal:

Labels of Default

Edit	Name	Color	Display format	Order [Bit/Byte]	Value
<input checked="" type="checkbox"/>	flags	■	Bit	MSB/BE	-
<input checked="" type="checkbox"/>	pressure	■	Bit	MSB/BE	-
<input checked="" type="checkbox"/>	temperature	■	Bit	MSB/BE	-
<input checked="" type="checkbox"/>	ID	■	Bit	MSB/BE	-
<input checked="" type="checkbox"/>	unknown	■	Bit	MSB/BE	-
<input checked="" type="checkbox"/>	CRC	■	Bit	MSB/BE	-

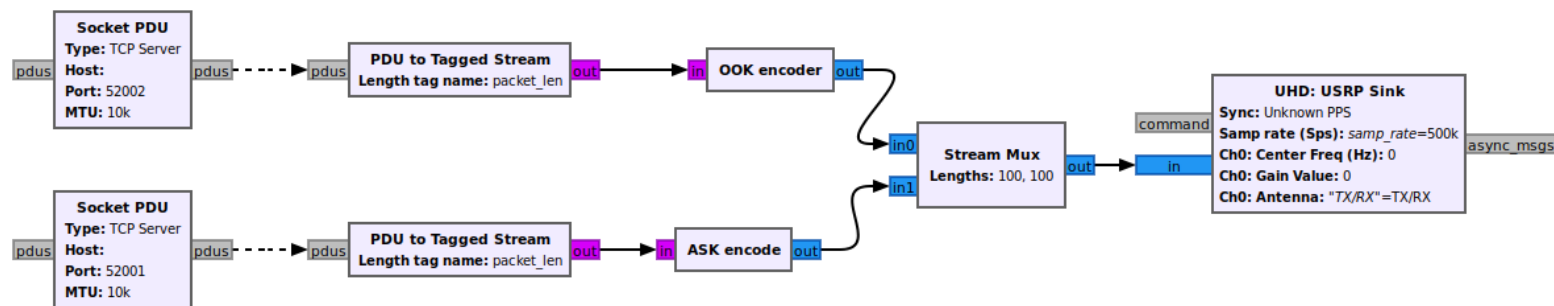
Transmission

- Requires a transmitter
- Raspberry Pi seems a cheap solution $\approx 50\text{€}$
- RPiTX allows transmitting over 5 KHz – 1.5 GHz



Handling two modulations

- Handling two modulations
- Hybrid sensors = more support?
- If only sending 1 modulated signal, we can also mix everything:

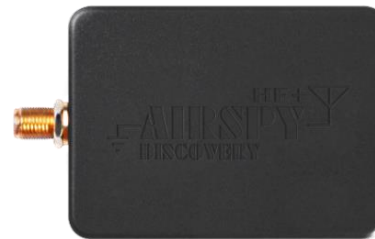


RF activation - captures

- A Low Frequency (LF) signal is sent → wake-up the sensor
- Frequency used: 125 KHz
- To receive it with RTL-SDR → need an up-converter (+50-100€)



- Or an AirSpy will do the job too



RF activation – transmission

- RPiTX supports 125 kHz theoretically
- Or use of USRP with DC-30 MHz daughterboard → much expen\$\$\$ive



LimeSDR specs says 100 kHz for TX → but need modifications

Risks

- Mostly Tracking
- Impersonating sensors → stopping the vehicle,
- or raising (crazy) notifications → driver in pain
- But not easy to trigger on the road:
 - Need to be in range, or transmit a signal with a decent gain → directional antenna + LNA

RDS and DAB/DAB+

Why do we care about Radio?

- AM and FM are just raw analogic signals → no structures
- But digital information carry:
 - Object types
 - Sometimes length of objects
 - Strings
 - ID
- There is maybe an area for fuzzing! ;)

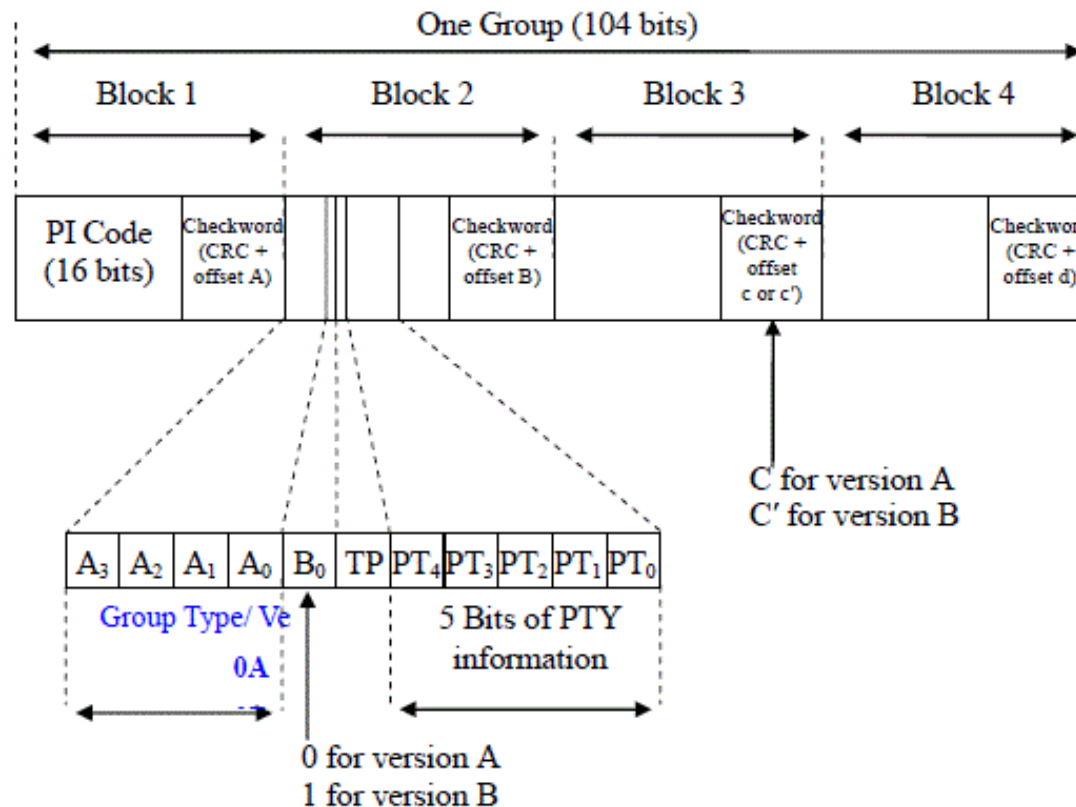
RDS

- Radio Data System (Radio Broadcast Data System (RBDS) for the U.S. version)
- Embeds digital information in FM radio broadcast
- Uses BPSK



RDS structure

- PI: Program ID code
- TP: Traffic Program code
- PTY: Program Type code
- **TA: Traffic Announcement**
- Etc.



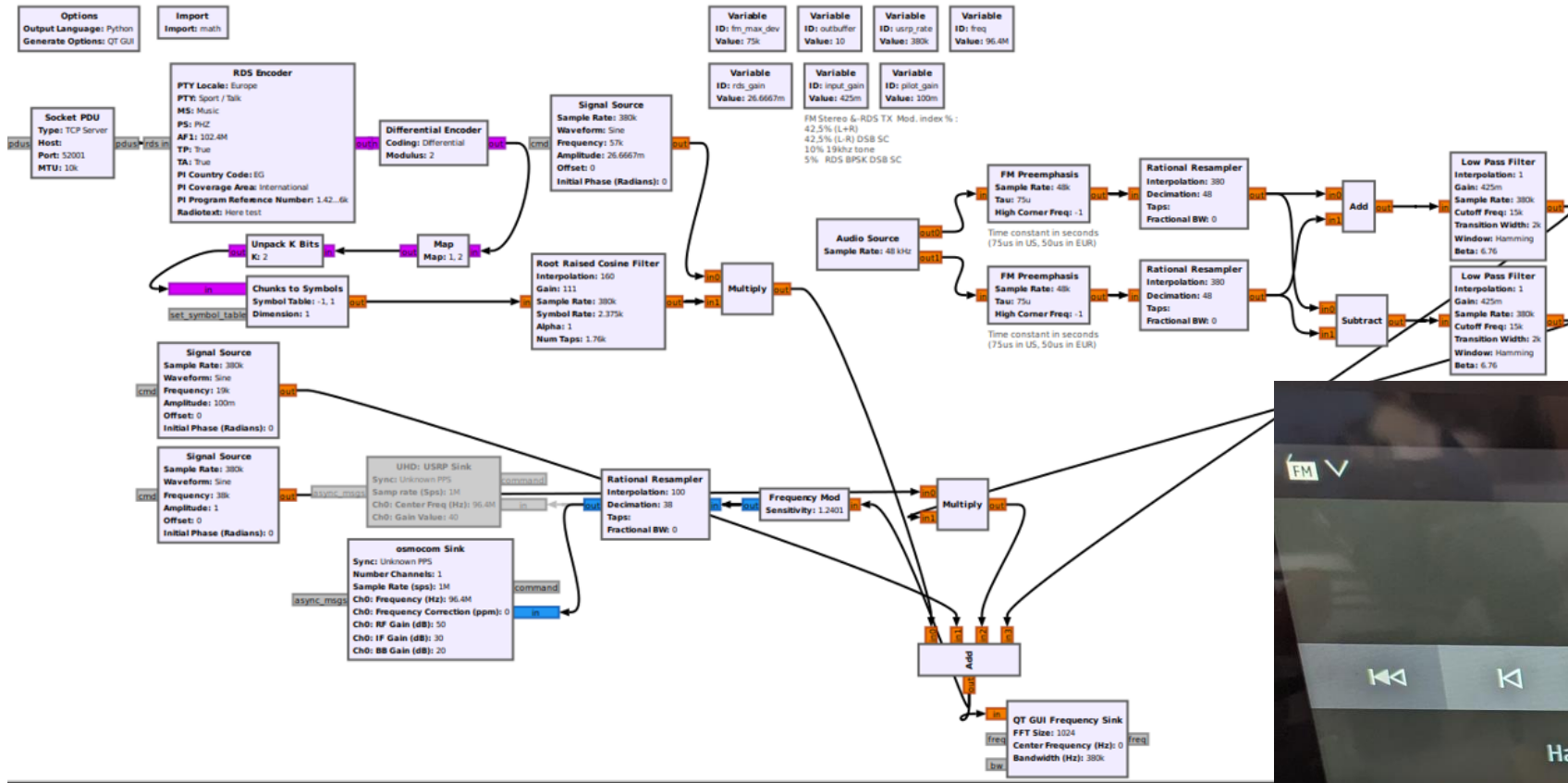
Go further by Friedt Jean-Michel: <https://connect.ed-diamond.com/GNU-Linux-Magazine/glmf-204/radio-data-system-rds-analyse-du-canal-numerique-transmis-par-les-stations-radio-fm-commerciales-introduction-aux-codes-correcteurs-d-erreur>

DAB

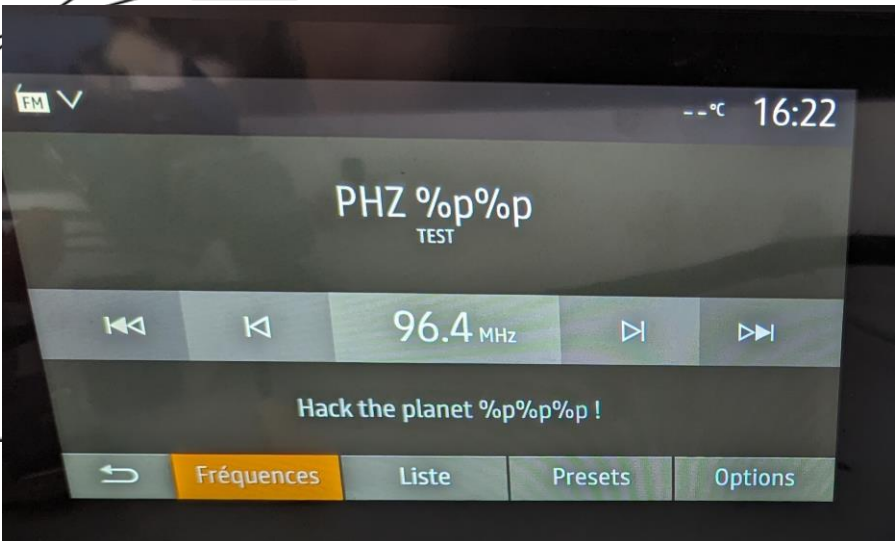
- Digital Audio Broadcasting
- DAB+ → upgrades for more stations with HD quality



RDS injections attempts

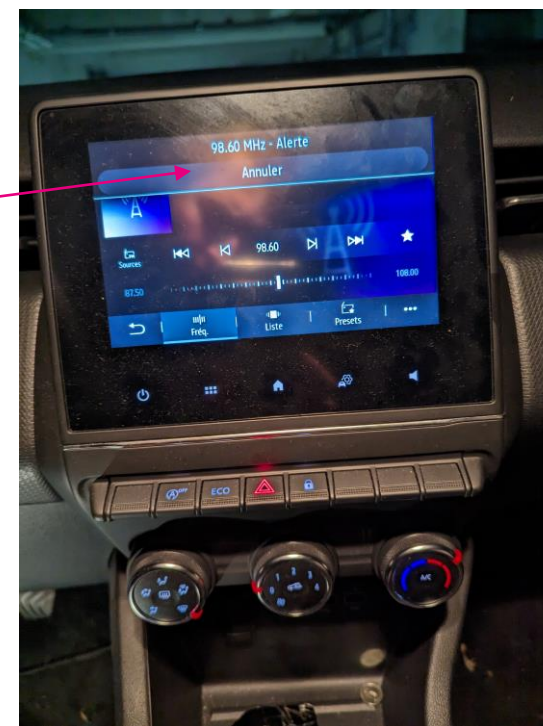


*with a modified gr-rds + libs ;)



RDS Alerts

- With a modified version → fuzzing:
 - PI
 - TP
 - PTY
 - Etc.
 - **But also TMC events 😊**



RDS Alerts

- Many events exist:

```
{ "1168", "security alert", "1515", " " },
{ "1169", "security incident", "1476", " " },
{ "1170", "police checkpoint", "1477", " " },
{ "1171", "bomb alert", "1516", " " },
{ "1172", "terrorist incident", "1478", " " },
{ "1173", "gunfire on roadway, danger", "1479", " " },
{ "1174", "civil emergency", "1480", " " },
{ "1175", "air raid, danger", "1481", " " },
{ "1176", "evacuation", "1494", " " },
{ "1177", " ", " ", " ", " " },
{ "1178", "air raid warning cancelled", "1587", " " },
{ "1179", "security alert withdrawn", "1492", " " },
{ "1180", "civil emergency cancelled", "1588", " " },
{ "1181", " ", " ", " ", " " }
```

- Warning → broadcasted → alerts all vehicles around → use carefully in a Faraday cage

DAB injection

- Same with a modified “DAB step”! :

The screenshot shows the 'DAB step' software interface. At the top left is the logo 'DAB step'. Below it are tabs for 'Receiver' and 'Transmitter', with 'Transmitter' selected. The interface is divided into several sections:

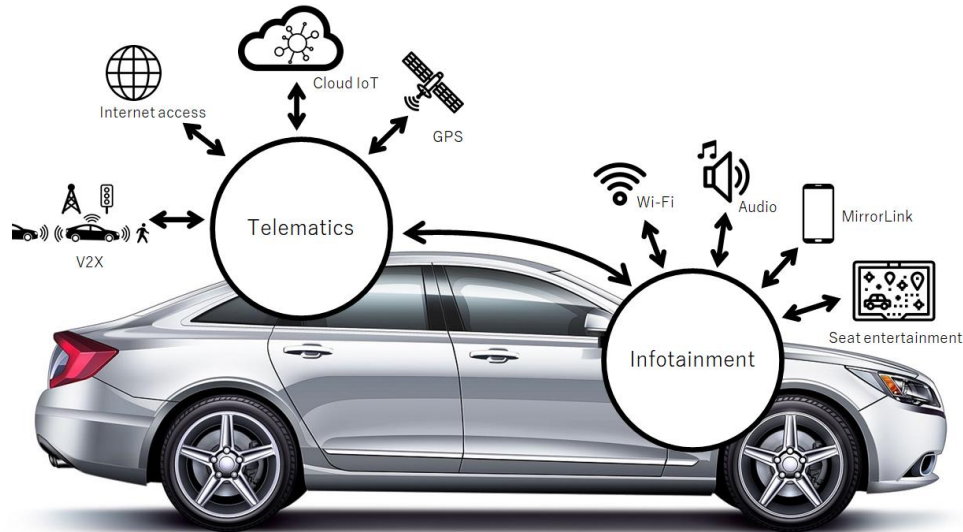
- Source Selection:** Radio buttons for 'USRP' (selected) and 'File' (with 'gen_iq_dab.dat' in the input field). A 'select path' button is next to the file input.
- Frequency:** A dropdown menu showing '201072000 Hz'.
- Transmission Mode:** A dropdown menu showing '1'.
- Service Components:** A section titled 'Component 1' with a 'DAB+' dropdown. It contains fields for 'Name', 'Data rate [kbit/s]' (set to 112), 'Protection Mode' (set to A1), 'Audio settings' (set to stereo and 32 kHz), and 'Audio Source' (with a 'select audio' button).
- Ensemble info:** A section with dropdown menus for 'Label' (PHZ <3), 'Country' (Germany), 'Number of channels' (1), and 'Language' (French).

A 'Developer Mode' button is visible in the top right corner.

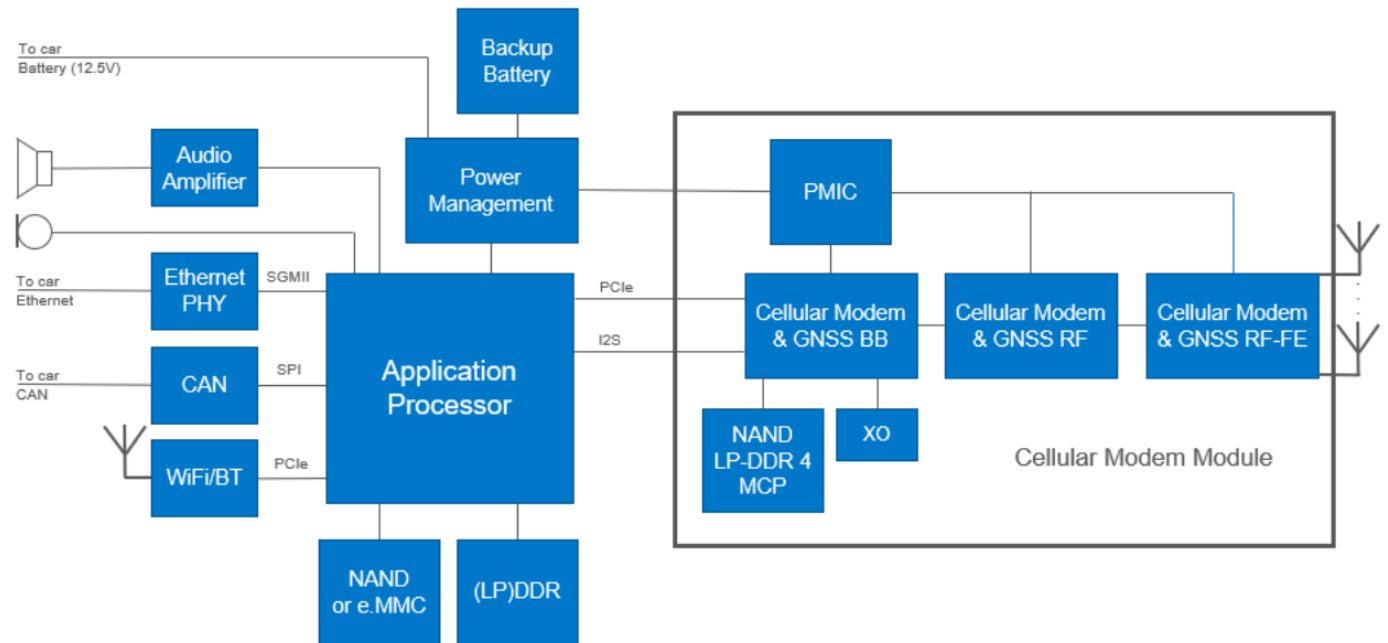
Mobile access on IVI

TCUs with 3G-5G stacks used in cars

5G → not very common, but starting to be developed



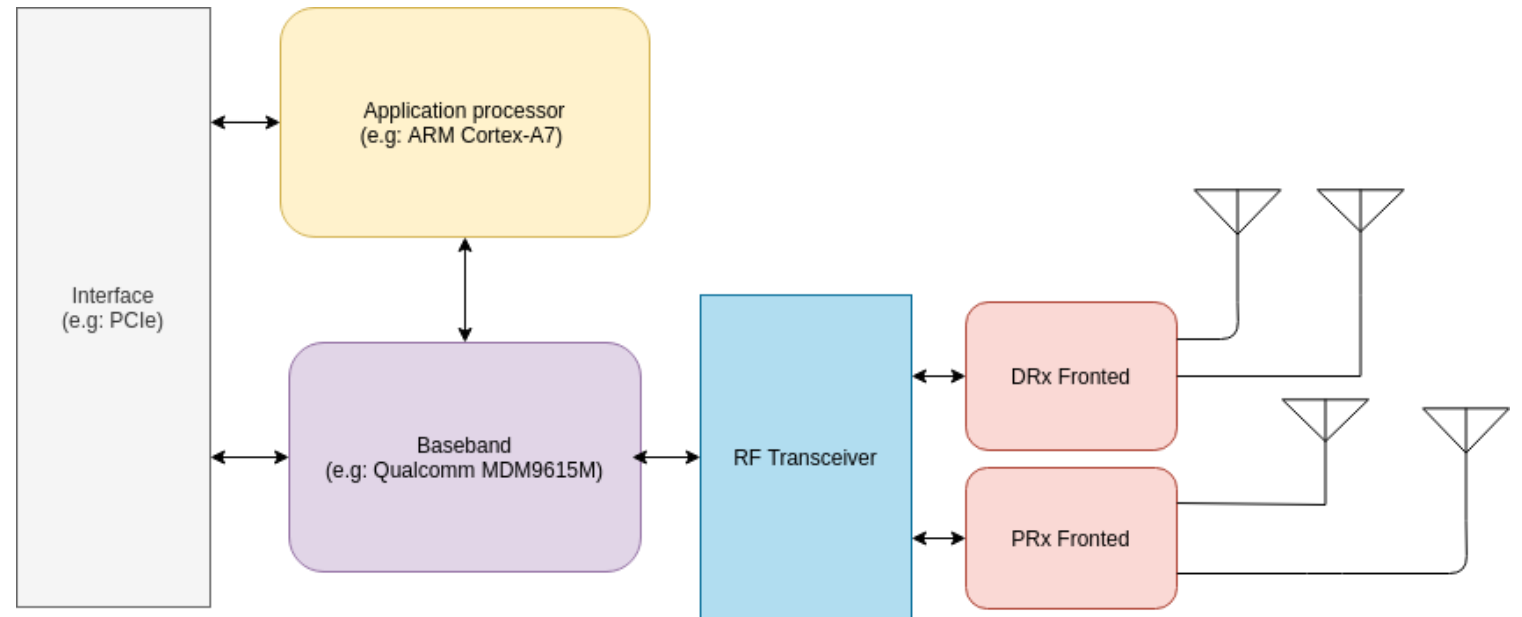
Source: <https://www.i-pex.com/>



Source: <https://media-www.micron.com/>

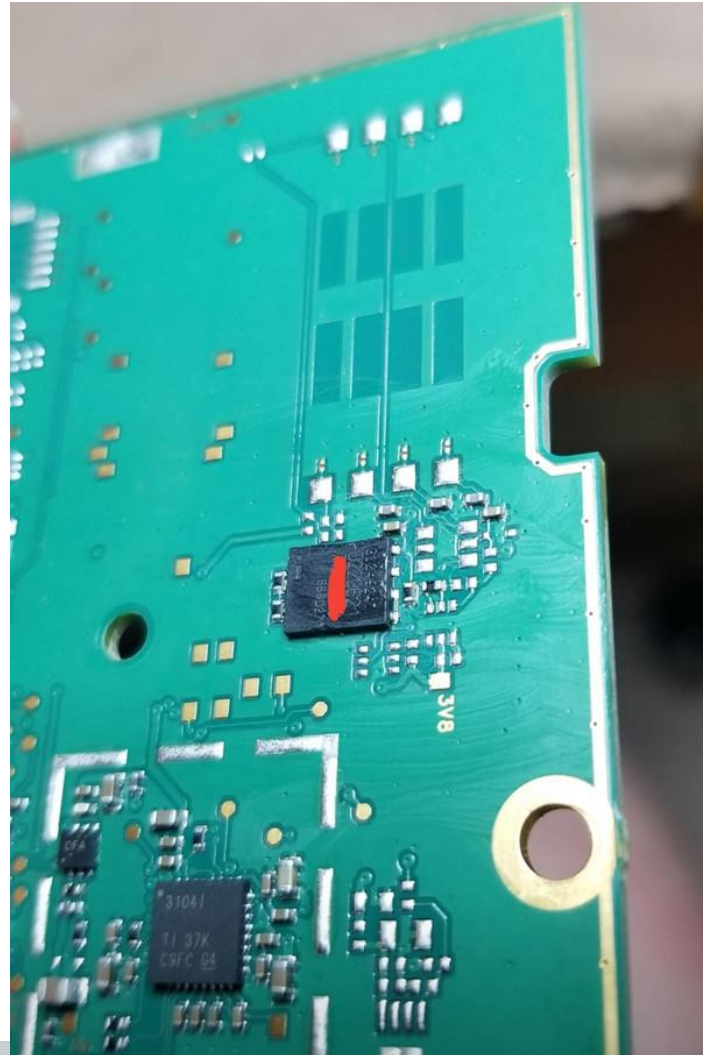
What do they have in common?

- Composed of:
 - Applicative processor
 - 2 frontends:
 - DRx & PRx → radio transmission
 - Baseband processor → implementing the mobile stacks
 - Memory:
 - NAND & DDR
 - And other interfaces...



Mobile access on IVI

Soldered eUICC



<https://f30.bimmerpost.com/forums/showthread.php?t=1642417>

Interception with soldered eUICC

- After desoldering, we can put our custom SIM card
- If IP is whitelisted, we can use the legitimate SIM card with a computer to forward accesses:



Soldered eUICC but extra SIM slot

- Embedded SIM needs to be chipped off before hooking them
- But 2nd slot exists in most cases + need to force the use with AT commands

Pin name	Pin no.	Electrical description	Description	Comment
(U)SIM1_PWR	36	PO	Power supply for (U)SIM1 card	
(U)SIM 1_DATA	34	DIO	(U)SIM1 card data, which has been pulled up to (U)SIM1_VDD via a 20KR resistor internally	
(U)SIM 1_CLK	32	DO	(U)SIM1 clock signal	
(U)SIM1_RESET	30	DO	(U)SIM1 Reset control	
(U)SIM 1_DET	66	DI	(U)SIM1 card detect, which has been pulled up to VDD_P3 via a 470KR resistor internally	1.8/3.0V voltage domain, all (U)SIM interfaces should be
(U)SIM2_PWR	48	PO	Power supply for (U)SIM2 card	protected against ESD.
(U)SIM2_DATA	42	DIO	(U)SIM2 card data, which has been pulled up to (U)SIM2_VDD via a 20KR resistor internally	If unused, please keep open
(U)SIM2_CLK	44	DO	(U)SIM2 clock signal	
(U)SIM2_RESET	46	DO	(U)SIM2 Reset control	
(U)SIM2_DET	40	DI	(U)SIM2 card detect, which has been pulled up to VDD_P3 via a 470KR resistor internally	

IVI and telematic systems in cars

- Usually use the mobile network:
 - Updates
 - Applications (Twitter, Facebook, etc.)
 - In-car internet
 - Streaming
 - Etc.
- Use GSM/GPRS, 3G, 4G stacks (soon 5G)

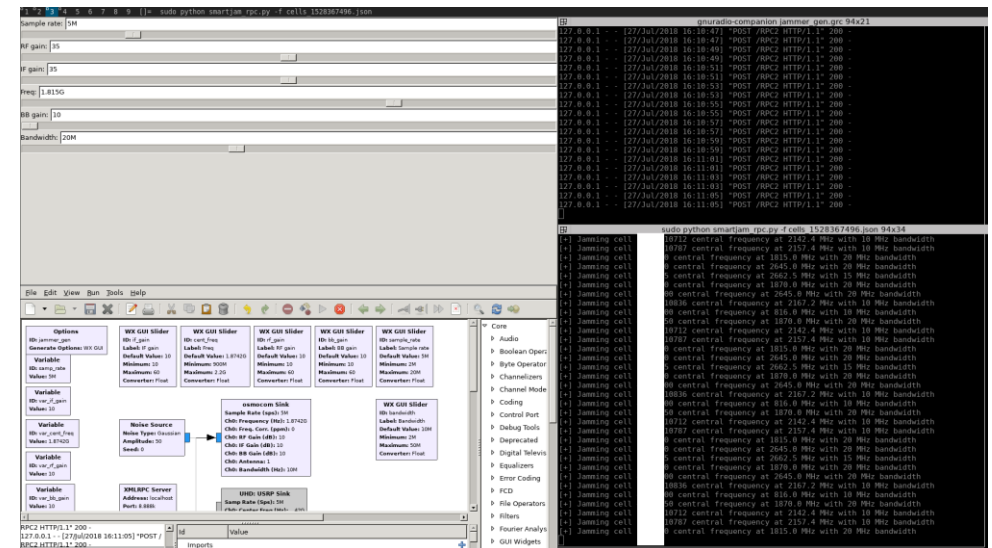
Possible attacks

- Eavesdropping in 2G:
 - no mutual authentication
 - A5/0 can be enforced
- Downgrading from 4G/3G to 2G:
 - Jamming
 - Parking places

Jamming

- Can use jammer (e.g from AliExpress)
- Or SDR to jam
- Smart jamming tools → Modmobjam

<https://github.com/PentHertz/Modmobjam>



Downgrading security: smart way

- Like for 4G, playing with Tracking Area Update procedure → reject causes → make the baseband switching to older stacks → need to modify srsRAN's stack
- **New:** 5G NSA NEA0 Bidding-Down Attack + 5G to 2G demonstration in “Never Let Me Down Again: Bidding-Down Attacks and Mitigations in 5G and 4G” by Bedran Karakoc, Nils Fürste, David Rupprecht, Katharina Kohls from Radix-security

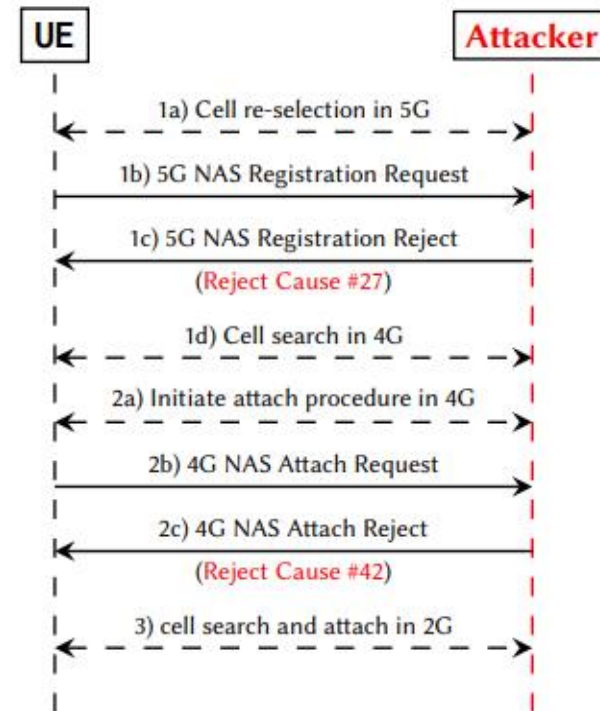
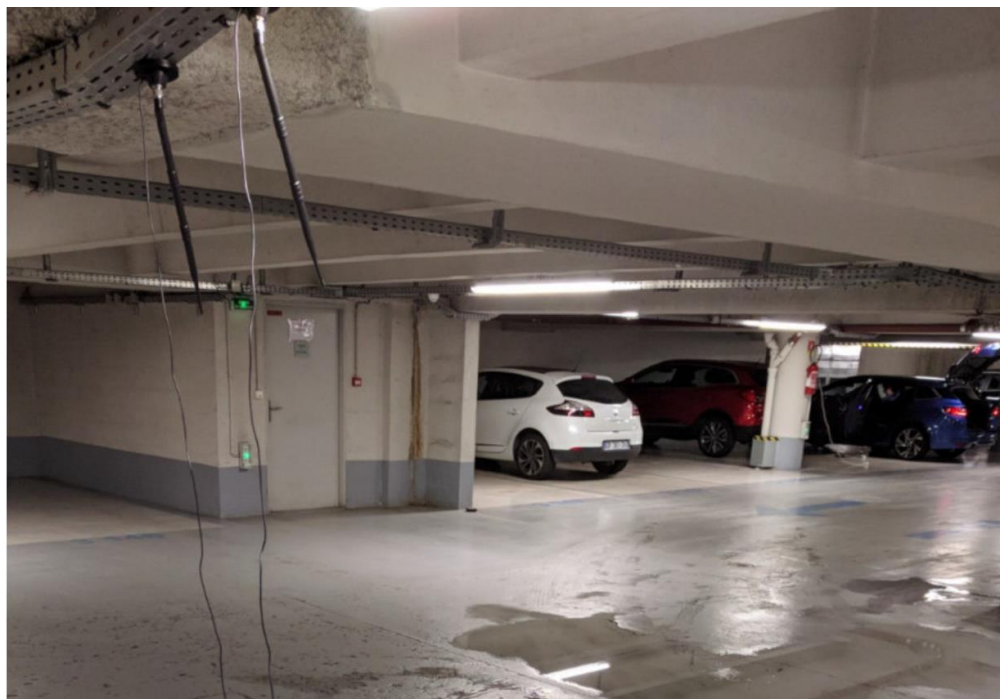


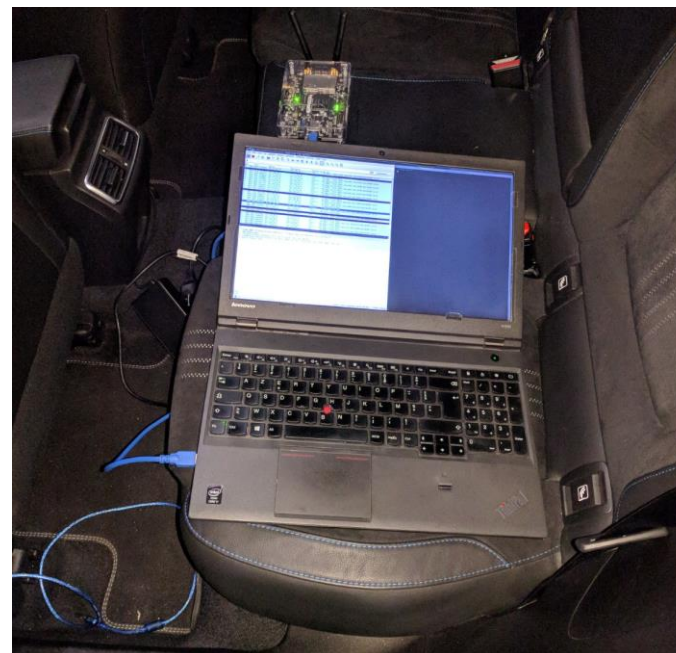
Figure 3: Protocol flow of downgrade dance from 5G to 2G.

Or good old parking places

- Sometimes good “Faraday cage”



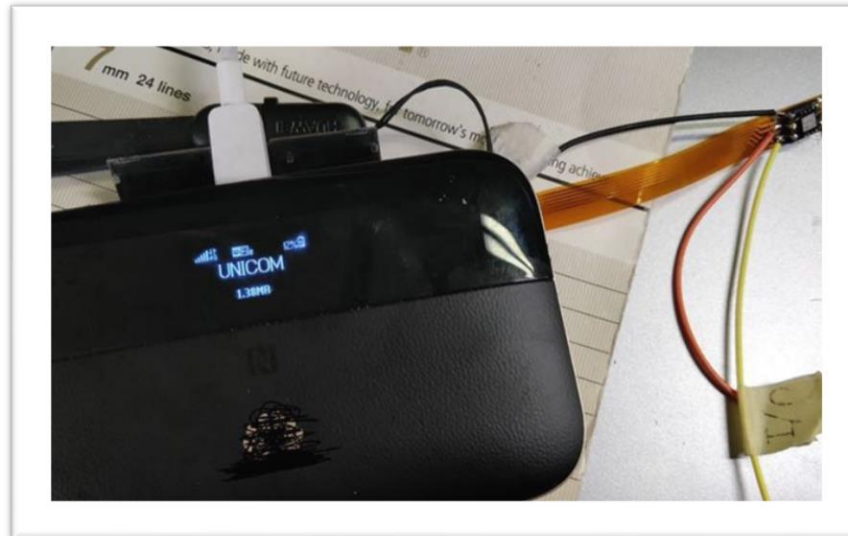
```
10 1.459318826 192.168.99.2 192.168.99.254 HTTP 913 POST /Service/InitSession/ HTTP/1.1 (applicat
19 7.536599505 192.168.99.2 10.91.80.203 HTTP 52 HEAD http://master.coyoterts.com HTTP/1.1
26 13.660617735 192.168.99.2 10.91.80.203 HTTP 52 HEAD http://master.coyoterts.com HTTP/1.1
65021 922.704281910 192.168.99.2 10.91.80.203 HTTP 52 HEAD http://master.coyoterts.com HTTP/1.1
66923 946.703883356 192.168.99.2 10.91.80.203 HTTP 52 HEAD http://master.coyoterts.com HTTP/1.1
69066 974.461373298 192.168.99.254 192.168.99.2 HTTP 173 HTTP/1.0 404 File not found
69093 974.818419668 192.168.99.2 192.168.99.254 HTTP 52 HEAD http://master.coyoterts.com HTTP/1.1
70396 990.503915759 192.168.99.2 192.168.99.254 HTTP 406 POST /api/app/call HTTP/1.1 (application/x-protobuf)
70401 990.504770592 192.168.99.254 192.168.99.2 HTTP 390 HTTP/1.0 501 Unsupported method ('POST') (text/html)
+ 70459 991.484062985 192.168.99.2 192.168.99.254 HTTP 406 POST /api/app/call HTTP/1.1 (application/x-protobuf)
70462 991.484923386 192.168.99.254 192.168.99.2 HTTP 390 HTTP/1.0 501 Unsupported method ('POST') (text/html)
70530 992.483719425 192.168.99.2 192.168.99.254 HTTP 406 POST /api/app/call HTTP/1.1 (application/x-protobuf)
70533 992.484544176 192.168.99.254 192.168.99.2 HTTP 390 HTTP/1.0 501 Unsupported method ('POST') (text/html)
1048... 1590.1445388... 192.168.99.2 192.168.99.254 HTTP 406 POST /api/app/call HTTP/1.1 (application/x-protobuf)
1048... 1590.1450970... 192.168.99.254 192.168.99.2 HTTP 390 HTTP/1.0 501 Unsupported method ('POST') (text/html)
1048... 1591.0455681... 192.168.99.2 192.168.99.254 HTTP 406 POST /api/app/call HTTP/1.1 (application/x-protobuf)
1048... 1591.0462935... 192.168.99.254 192.168.99.2 HTTP 390 HTTP/1.0 501 Unsupported method ('POST') (text/html)
1049... 1591.8855224... 192.168.99.2 192.168.99.254 HTTP 406 POST /api/app/call HTTP/1.1 (application/x-protobuf)
```



- Old Android are used → choice of RCE

Going further on the backend

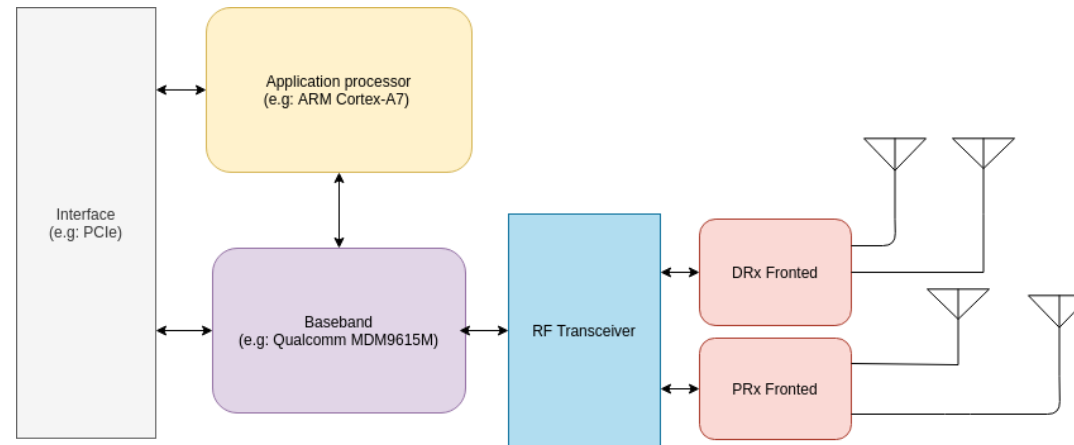
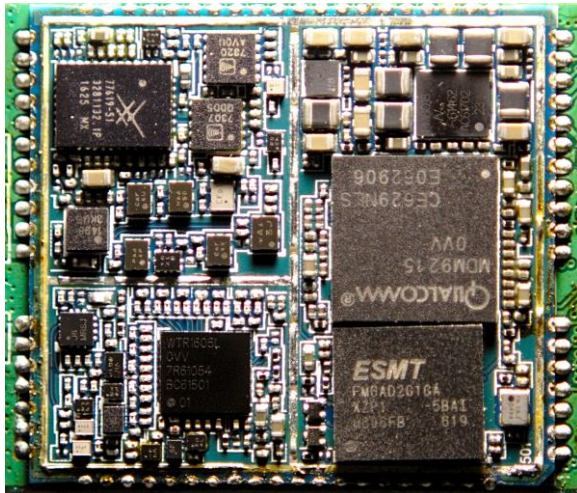
- We can try attacking the backend
- We can extract the eSIM and readapt pins for a modem:



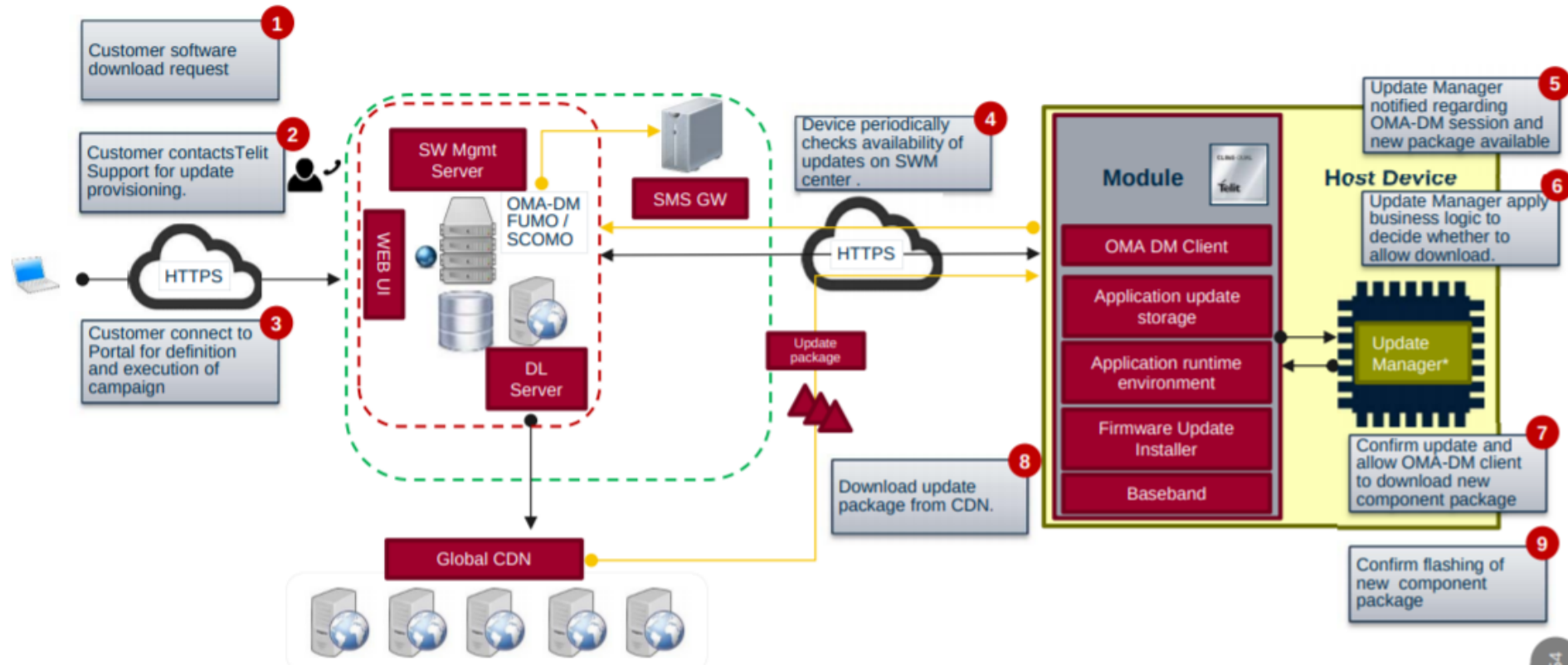
Source: Security Research on Mercedes-Benz: From Hardware to Car Control by Minrui Yan, Jiahao Li and Guy Harpak

Mobile modules

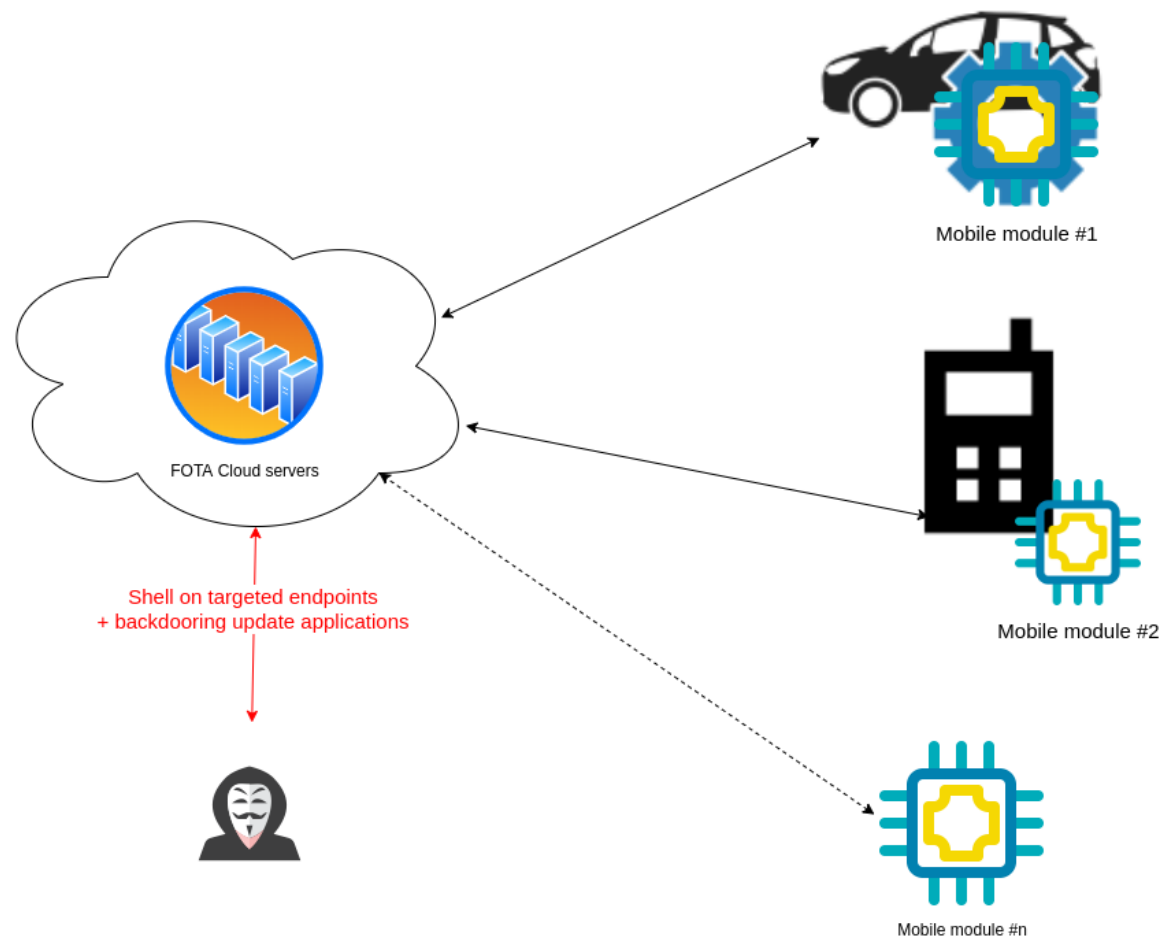
- Used in IoT and cars to communicate with the mobile network



FOTA updates: schema



FOTA updates: Impacts





**Some nice
opportunities**

Getting an RCE in a car

- Hard ways → exploiting a corruption → but need a context
- Smart ways → Finding a service exposed in one of the interfaces:
 - Wi-Fi → sometimes needs to intercept BT traffic → Wi-Fi password
 - Mobile
 - Ethernet direct or USB OTG

Recurrent candidates

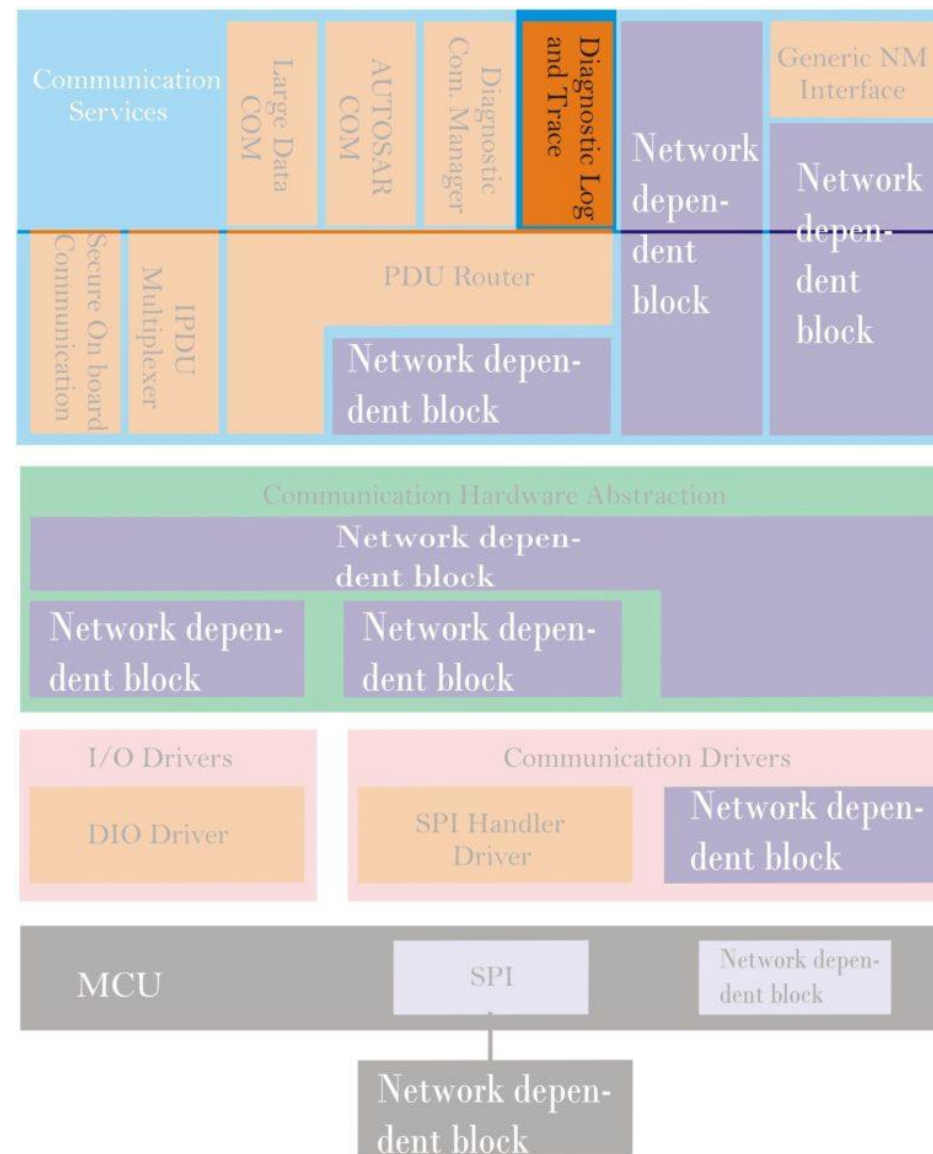
- QNX in uses:
 - Look at exposed **qconn** service 😊 (good old trick! But with a little update)

```
user@testlab:~$ telnet
telnet> open 192.168.86.125 8000 # target's IP address
Trying 192.168.86.125...
Connected to 192.168.86.125.
Escape character is '^]'.
QCONN
<qconn-broker> service launcher
OK
<qconn-launcher> start/flags run /sbin/shutdown -b
OK 970775
^[[3~^M^MConnection closed by foreign host.
```


Some nice opportunities

DLT?

- Diagnostic Log and Trace
- Sender-receiver communication
- See more: <https://autosartutorials.com/diagnostic-log-and-trace/>



Some nice opportunities

DLT traces

- Can trace:
 - Events
 - Crashes
 - Running processes

Index	Time	Timestamp	Ecuid	Apid	Ctid	Type	Payload		
1615	202	5.5... 936.2404		↓	RADI	RADI	log	void	uint16, const QString&, int, bool) UpdatePresetView: freq 9840 PI 65158 PSN YVELINES
1616	202	5.5... 936.2405		↓	RADI	RADI	log	QList	:hannelsInPreset(uint16, quint16, bool) FindChannelsInPreset: Freq = 9840 - PI = 65158 - bRdsEnabled = 1 - AF of
1617	202	5.5... 936.2406		↓	RADI	RADI	log	void	(bool) Set RemovePresetByPIUpdated: 1 -> 1
1618	202	5.5... 936.2406		↓	RADI	RADI	log	QList	:hannelsInPreset(uint16, quint16, bool) FindChannelsInPreset: Freq = 9840 - PI = 65158 - bRdsEnabled = 1 - AF of
1619	202	5.5... 936.2407		↓	RADI	RADI	log	void	uint16, const QString&, int, bool) Not found match item preset
1620	202	5.5... 936.2407		↓	RADI	RADI	log	void	, bool) YVELINES
1621	202	5.5... 936.2407		↓	RADI	RADI	log	void	, bool) Delay 200ms to sending media info
1622	202	5.5... 936.2408		↓	RADI	RADI	log	void	FM: YVELINES -> 98.4 FM
1623	202	5.5... 936.2409		↓	RADI	RADI	log	Radic	uint16, quint16, bool) FindChannelsInList: Freq = 9840 - PI = 65158 - bRdsEnabled = 1 - AF opt = 1
1624	202	5.5... 936.2409		↓	RADI	RADI	log	void	uint16, const QString&, int, bool) UpdatePresetView: freq 9840 PI 65158 PSN 98.4 FM
1625	202	5.5... 936.2411		↓	RADI	RADI	log	QList	:hannelsInPreset(uint16, quint16, bool) FindChannelsInPreset: Freq = 9840 - PI = 65158 - bRdsEnabled = 1 - AF of
1626	202	5.5... 936.2411		↓	RADI	RADI	log	void	, bool) 98.4 FM
1627	202	5.5... 936.2412		↓	RADI	RADI	log	void	, bool) Last media info sending was not finished for 200ms, wait
1628	202	5.5... 936.2542		↓	MM...	MM...	log	Micor	d9 00 7d f6 00 00
1629	202	5.6... 936.3758		↓			control	[]	
1630	202	5.7... 936.4209		↓	MM...	MM...	log	READ	
1631	202	5.7... 936.4211		↓	MM...	MM...	log	Radic	
1632	202	5.7... 936.4315		↓	RADI	RADI	log	void	rel info: {"info":"98.4 FM","launch":"com.lge.bavn.appradio","source":"Radio"}
1633	202	5.7... 936.4335		↓	HO...	INFO	log	[boo	String&, const QString&) isEnabled: 1 ~ ~ source_audio: FM ~ ~ name_played: 98.4 FM
1634	202	5.7... 936.4338		↓	HO...	INFO	log	[boo	String&, const QString&) m_listActiveAudioSource: count= 1
1635	202	5.7... 936.4339		↓	HO...	INFO	log	[boo	String&, const QString&) PopupSystem is displayed, save data to cache!!!
1636	202	5.7... 936.4363		↓	RADI	RADI	log	void	DBusPendingCallWatcher*) Send channel info to home screen successfully
1637	202	5.7... 936.4374		↓	MIPV	MIPC	log	hand	nfo":"98.4 FM","launch":"com.lge.bavn.appradio","source":"Radio"}
1638	202	5.7... 936.4375		↓	MIPV	MIPC	log	sendl	h":"com.lge.bavn.appradio","source":"Radio"}
1639	202	5.7... 936.4375		↓	MIPV	MIPC	log	Medi	unch":"com.lge.bavn.appradio","source":"Radio"}
1640	202	5.7... 936.4411		↓	RADI	RADI	log	void	DBusPendingCallWatcher*) Send channel info to navi successfully
1641	202	5.8... 936.5810		↓	MM...	MM...	log	READ(23) a	

- Perfect to debug fuzzing when it's exposed! 😊

DLT RCE?

- Interesting function:
 - Possible to reach with right ECU ID + Service ID if the configuration allows!

```
dlt-daemon / src / system / dlt-system-shell.c
Code Blame 121 lines (106 loc) · 4.82 KB
87         DLT_STRING("dlt-system-shell, injection data:");
88         DLT_STRING(text));
89
90         switch (service_id) {
91         case 0x1001:
92
93             if ((syserr = system(text)) != 0)
94                 DLT_LOG(shellContext, DLT_LOG_ERROR,
95                     DLT_STRING("dlt-system-shell, abnormal exit status."),
96                     DLT_STRING(text),
97                     DLT_INT(syserr));
98             else
99                 DLT_LOG(shellContext, DLT_LOG_INFO,
100                     DLT_STRING("Shell command executed:"),
101                     DLT_STRING(text));
102
103             break;
104         default:
105             DLT_LOG(shellContext, DLT_LOG_ERROR,
106                 DLT_STRING("dlt-system-shell, unknown command received."),
107                 DLT_UINT32(service_id),
108                 DLT_STRING(text));
109             break;
110         }
111
```

RKE systems

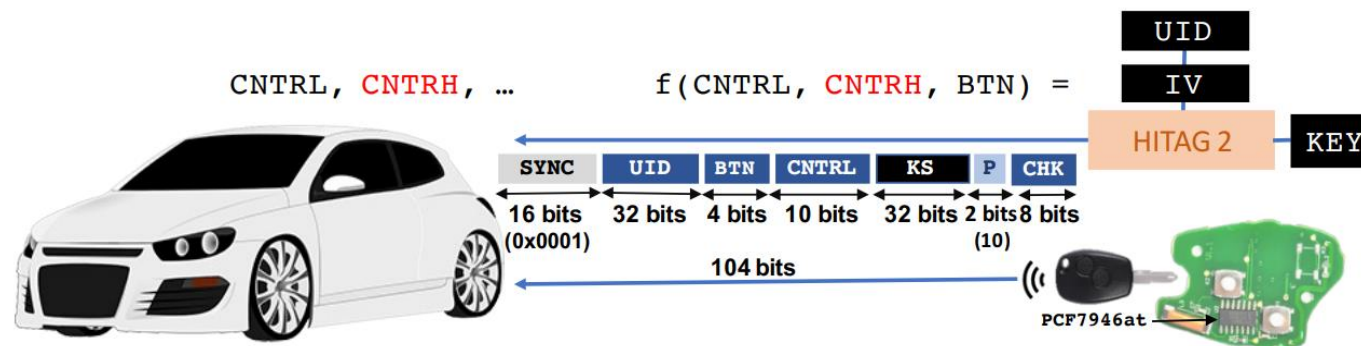
Practical attack on RKE with Hitag2

- Remote keyless Entry
- Different modes:
 - Fixed code → old & rare today
 - Rolling code
 - IFF (Identify Friend or Foe)



Practical attack on RKE with Hitag2

- Secrets are needed to perform efficient bruteforce
- Possible to get the shared static key out of the PCF7946



Source: From Academia to Real World : a Practical Guide to Hitag-2 RKE System
Analysis by Ryad Benadjila, Mathieu Renard, José Lopes-Esteves, Chaouki Kasmi

*Internet has some nice bruteforcers code, even in GPU :)

RKE vs Rollback attacks

- Sometimes replaying old consecutive code → roll back

Rolling Pwn Attack



INTRODUCTION

Modern vehicles are often equipped with a remote keyless entry system. These RKE systems allow unlocking or starting the vehicle remotely. The goal of our research was to evaluate the resistance of a modern-day RKE system. Our research disclosed a Rolling-PWN attack vulnerability affecting all Honda vehicles currently existing on the market (From the Year 2012 up to the Year 2022). This weakness allows anyone to permanently open the car door or even start the car engine from a long distance.

The Rolling-PWN bug is a serious vulnerability. We found it in a vulnerable version of the rolling codes mechanism, which is implemented in huge amounts of Honda vehicles. A rolling code system in keyless entry systems is to prevent replay attack. After each keyfob button pressed the rolling codes synchronizing counter is increased. However, the vehicle receiver will accept a sliding window of codes, to avoid accidental key pressed by design. By sending the commands in a consecutive sequence to the Honda vehicles, it will be resynchronizing the counter. Once counter resynced, commands from the previous cycle of the counter worked again.

Car apps

- Sometimes simpler than cracking RKEs hacking around Object IDs:
 - Remotely flashing the victim's vehicle's headlights
 - Honking the horn
 - Starting or stopping the engine
 - Locking or unlocking the car
 - Changing a PIN
 - Unlocking the boot

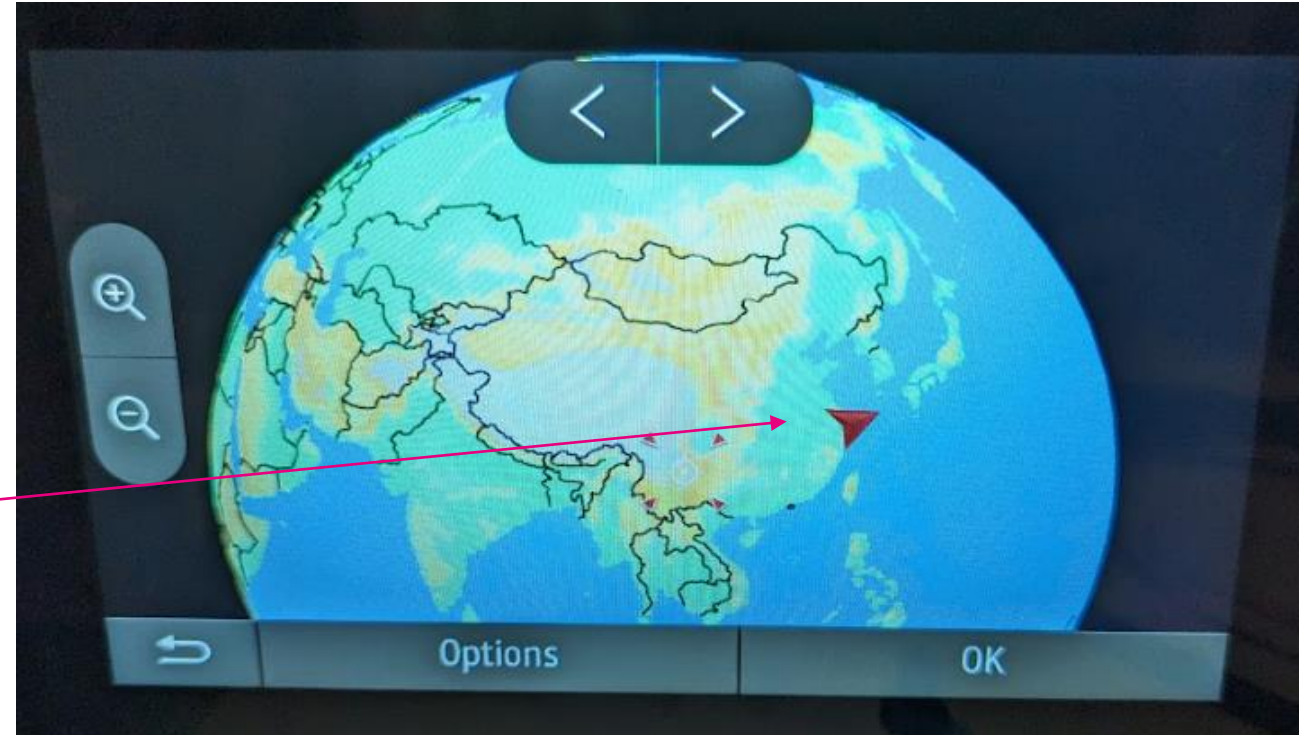




**GPS against
autopilot?**

Hijacking in action

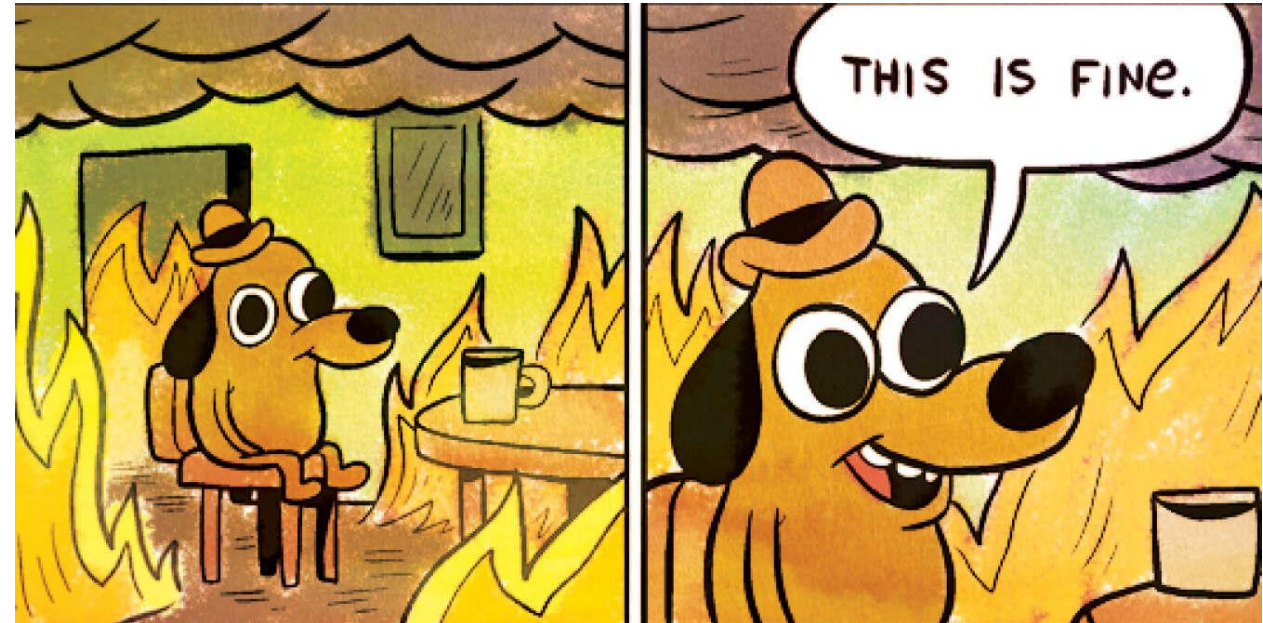
- The signal GPS can be hijacked
- Some GPS receiver look at how strong the signal is + other mechanisms to avoid this
- But doing that in the right way, it's still possible to teleport!



GPS against autopilot?

Hijacking vs Autopilot

- Question: What about Autopilot?

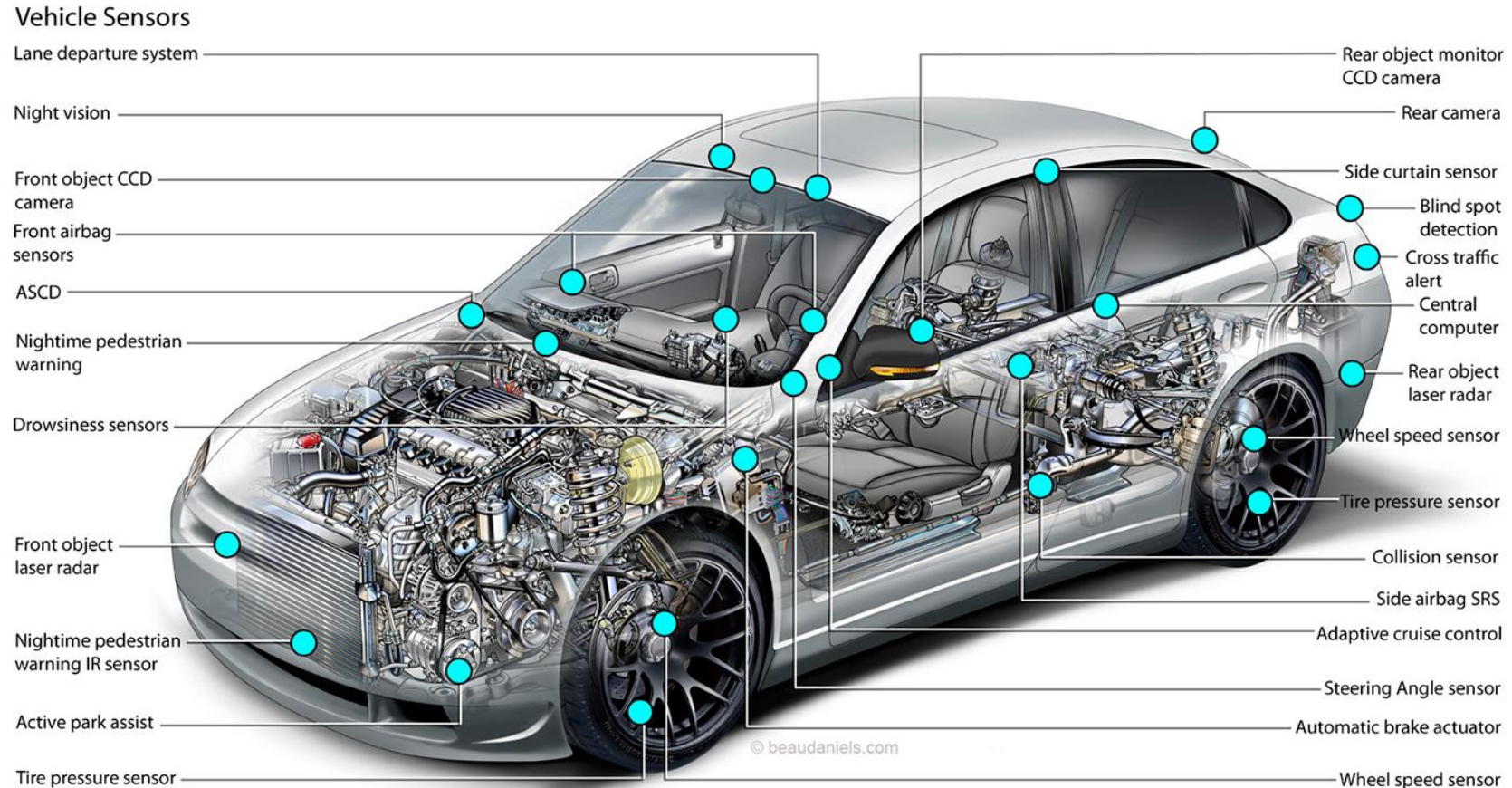




Going further

Going further

Sensors in cars > ~200

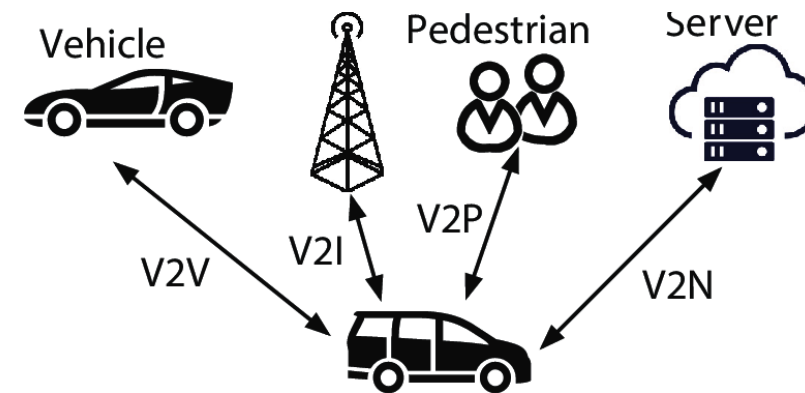


Vehicle Sensors, the connected car - <https://www.behance.net/gallery/51718817/Connected-car>

Going further

V2X

- Vehicle-to-everything
- For autonomous driving → safety, efficiency, and comfort
- C-ITS (Cooperative Intelligent Transport Systems) → standardize Connected Automated Driving (CAD)
- Type of communications →
 - V2I
 - V2N
 - V2V
 - V2P
 - V2D
- 802.11p → first deployed



Source: An Overview of 3GPP Cellular Vehicle-to-Everything Standards by Xuyu Wang, Shiwen Mao, Michelle X. Gong

Going further

802.11p

- Based on Wi-Fi
- DSRC in US
- ITS-G5 in EU
- But deployed first with some security concerns:
 - No privacy
 - No impersonation mechanism

Capturing 802.11p data

- More 2 ways:
 - Using a dedicated dongle with a modified kernel[1]
 - Using and adapting Openwifi projects [2], or bladerf-wiphy[3]
 - Or still using at least a USRP B with WIME (allows also TX!):

The screenshot displays the Wireshark network protocol analyzer interface. On the left, the 'Scope Plot' shows a constellation diagram with a grid of blue dots, indicating the modulation scheme. The plot axes are labeled 'ch1' and 'ch2'. The main window shows a list of captured packets, with packet 14 selected. The packet details pane on the right shows the structure of the selected packet: IEEE 802.11 QoS Data, Logical-Link Control, and Data (562 bytes). The packet bytes pane shows the raw hexadecimal and ASCII data. The status bar at the bottom indicates 26 packets displayed (100.0%) and a load time of 0:0.

- [1] <https://harrisonsand.com/posts/802-11p-v2x-hunting/>
- [2] <https://github.com/open-sdr/openwifi>
- [3] <https://www.nuand.com/bladerf-wiphy/>

Example of a capture: CAMv1 message

No.	Time	Source	Destination	Protocol	Length	AMF	Info
13	12.001925	0.15.0.04:e5:48:dc:...	Broadcast	CAMv1	146		CAMv1
14	12.002012	0.15.0.04:e5:48:dc:...	Broadcast	CAMv1	146		CAMv1


```
▼ GN_ADDR: 3c0004e548dca6e8
  0... .. = Manual: 0
  .011 11.. = ITS-S type: Road Side Unit (15)
  .... ..00 0000 0000 = ITS-S Country Code: Reserved (0)
  MID: CohdaWir_dc:a6:e8 (04:e5:48:dc:a6:e8)
  Timestamp: 3890296559ms
  Latitude: 59d55'17,77"N
  Longitude: 10d48'23,31"E
  1... .. = Position accuracy indicator: 1
  Speed: 327,68 m/s
  Heading: 0,0 degrees
  Reserved: 0
▼ BTP-B
  Destination Port: 2001
  Destination Port info: 0x0000
▼ Intelligent Transport Systems
  ▼ ItsPduHeader
    protocolVersion: 1
    messageID: cam (2)
    stationID: 184
  ▼ CoopAwarenessV1
    generationDeltaTime: Unknown (14337)
  ▼ camParameters
    ▼ basicContainer
      stationType: roadSideUnit (15)
      ▼ referencePosition
        latitude: Unknown (599278514)
        longitude: Unknown (108064763)
      ▼ positionConfidenceEllipse
        semiMajorConfidence: Unknown (95)
        semiMinorConfidence: Unknown (95)
        semiMajorOrientation: wgs84North (0)
      ▼ altitude
        altitudeValue: Unknown (614390)
        altitudeConfidence: alt-200-00 (13)
    ▼ highFrequencyContainer: rsuContainerHighFrequency (1)
      rsuContainerHighFrequency
```


0040	aa aa 03 00 00 00 89 47 11 00 50 01 20 50 02 00G..P..P..
0050	00 1e 01 00 3c 00 04 e5 48 dc a6 e8 e7 e1 36 ef	...<...H.....6..
0060	23 b8 43 b2 06 70 ef fb 80 00 00 00 00 00 00 00	#.C..p.....
0070	07 d1 00 00 01 02 00 00 00 b8 38 01 00 fb 2b a58...+..
0080	96 4e 37 58 3f 60 be 0b e0 01 5c d2 da 80 7c 2b	.N7X?`... ..\.... +
0090	ff cf	..

C-V2X

- Cellular V2X → LTE-V2X for the moment
- 2 modes of communications: Direct short-range & Network
- Powerful alternative to 802.11p (but 802.11bd is on its way!)
- Introduction of ProSe (Proximity Service) → Side Link → PC5 interface
- Defined by 3GPP
 - LTE: Rel. 12 & Rel. 13 → D2D and eD2D → Hazard warning
 - LTE Basic V2X by Rel. 14 → safety use case
 - 3GPP Release 15 → enhanced V2X → Enhanced Navigation & Infotainment
 - and 3GPP Release 16 includes work on 5G-NR → Cooperative auto. driving
- Current problem to solve → privacy protection and usurpation → use of PKI → handled by ETSI only not 3GPP

C-V2X tools and limitations

- LTE C-V2X implemented to srsRAN
 - Examples for Side Link RX
- Side Link → direct communication over PC5
- No SDR C-V2X for 5G-NR yet

```
[Length of frame: 233]
[Uplink grant size: 233]
[CRC Status: OK (1)]
[Carrier Id: Primary (0)]
MAC PDU Header (SL-SCH) (3:193) (Padding:remainder) [2 subheaders]
  Sub-header (SL-SCH)
    0011 .... = Version: 3
    .... 0000 = Reserved bits: 0x0
    Source Layer-2 ID: 0x72e066
    Destination Layer-2 ID: 0xaaaaaa
  Sub-header (lcid=3, length=193)
    00.. .... = Reserved bits: 0x0
    ..1. .... = Extension: 0x1
    ...0 0011 = LCID: 3 (0x03)
    1... .... = Format: Data length is >= 128 bytes
    .000 0000 1100 0001 = Length: 193
  Sub-header (lcid=Padding, length is remainder)
    00.. .... = Reserved bits: 0x0
    ..0. .... = Extension: 0x0
    ...1 1111 = LCID: Padding (0x1f)
SDU (3, length=193 bytes): 1860000000000510104e93c31353733373430333935393836...
Padding data: 00000000000000000000000000000000000000000000000000000000...
[Padding length: 29]

0000 01 00 08 02 10 01 03 00 01 04 00 00 07 01 0a 00 .....
0010 0f 00 01 30 72 e0 66 aa aa aa 23 80 c1 1f 18 60 ...Or.f...#...
0020 00 00 00 00 51 01 04 e9 3c 31 35 37 33 37 34 30 ....Q... <1573740
0030 33 39 35 39 38 36 30 30 31 3e 41 42 43 44 45 46 39598600 1>ABCDEF
0040 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 GHIJKLMN OPQRSTUV
0050 57 58 59 5a 41 42 43 44 45 46 47 48 49 4a 4b 4c WXYZABCD EFGHIJKL
0060 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 5a 41 42 MNOPQRST UVWXYZAB
0070 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 CDEFGHIJ KLMNOPQR
0080 53 54 55 56 57 58 59 5a 41 42 43 44 45 46 47 48 STUVWXYZ ABCDEFGH
0090 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 IJKLMN OPQRSTUVWX
00a0 59 5a 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e YZABCDEF GHIJKLMN
00b0 4f 50 51 52 53 54 55 56 57 58 59 5a 41 42 43 44 OPQRSTUV WXYZABCD
00c0 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 EFGHIJKL MNOPQRST
00d0 55 56 57 58 59 5a 41 42 43 44 45 46 47 48 49 00 UVWXYZAB CDEFGHI-
00e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

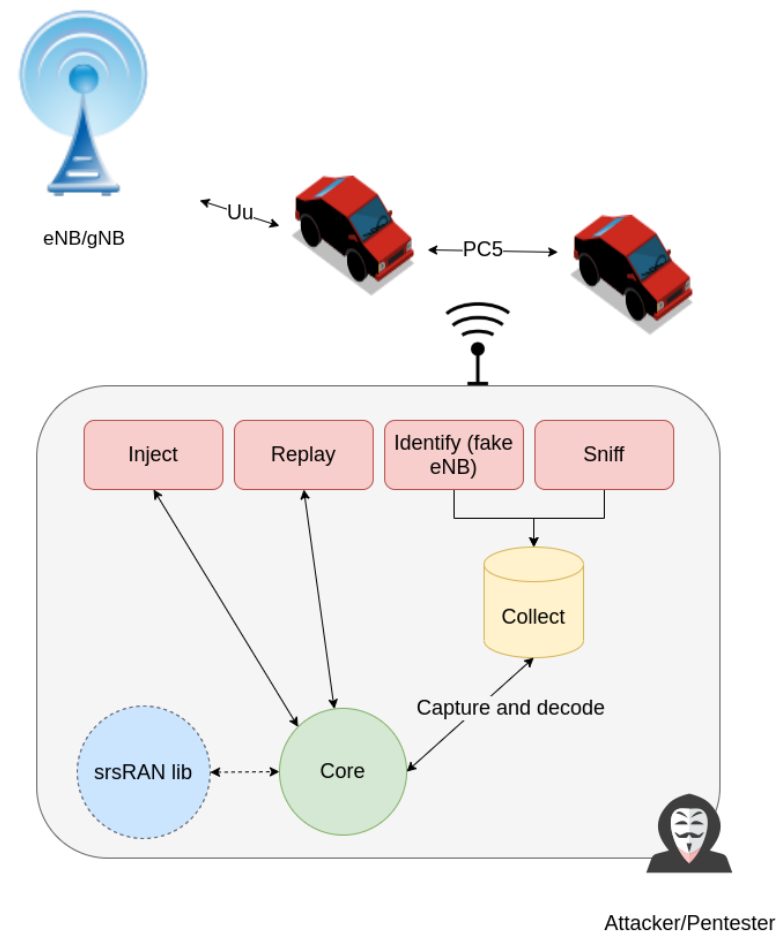
Going further

Our tool on LTE-V2X

- Based band srsRAN
- Focuses on PC5 mode 4
- Features:
 - Detection of capable V2X devices
 - Intercept and inspect SL messages
 - Injection of messages in current dev.

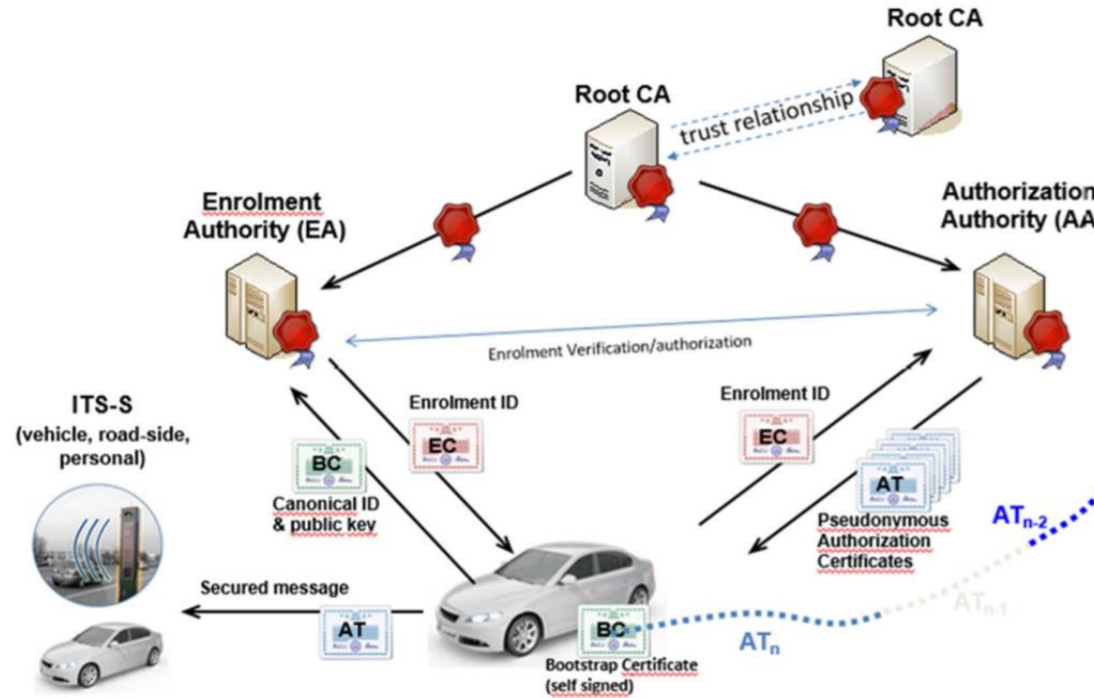
Current state of this research: Payed for some RSUs + OBUs kits during chip shortage → got scammed or getting those kits are still complicated after some years...

So we have only a simulator working, no real products to test ☹️



Going further

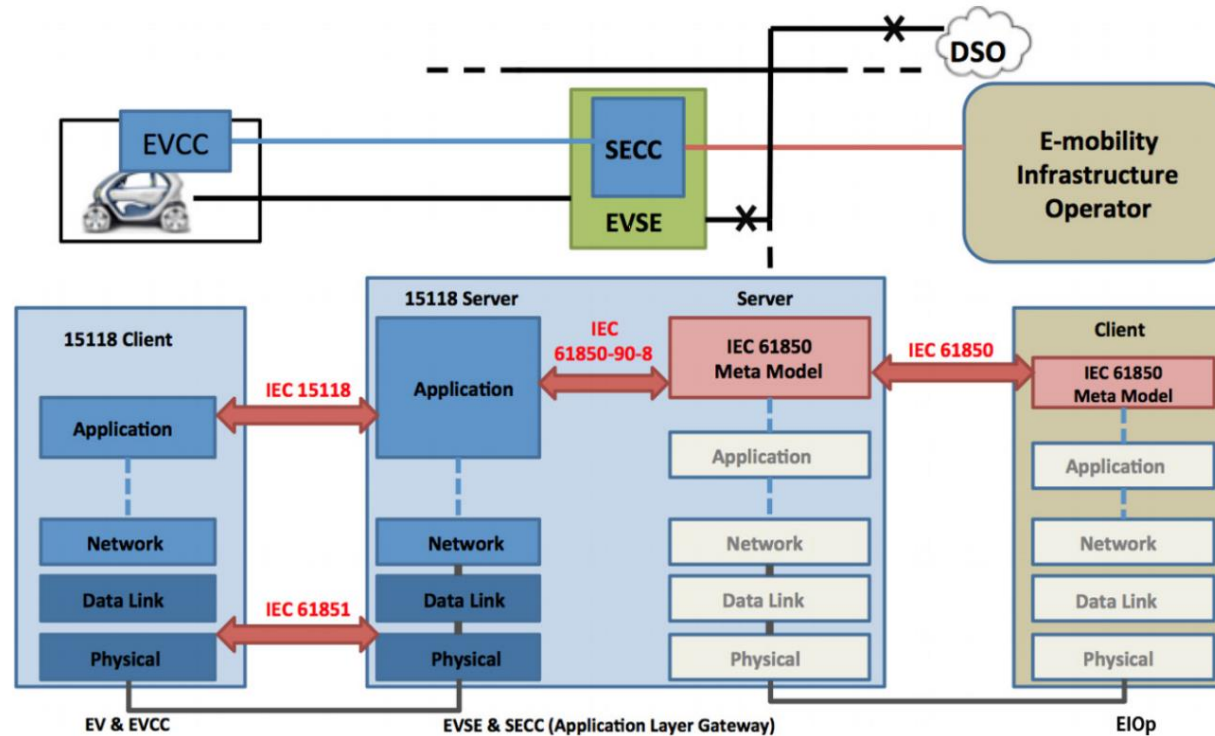
V2V/V2I PKI: What is the real state?



Source: ETSI TR 103 415 V1.1.1 (2018-04)

Going further

PKI: Remember V2G?



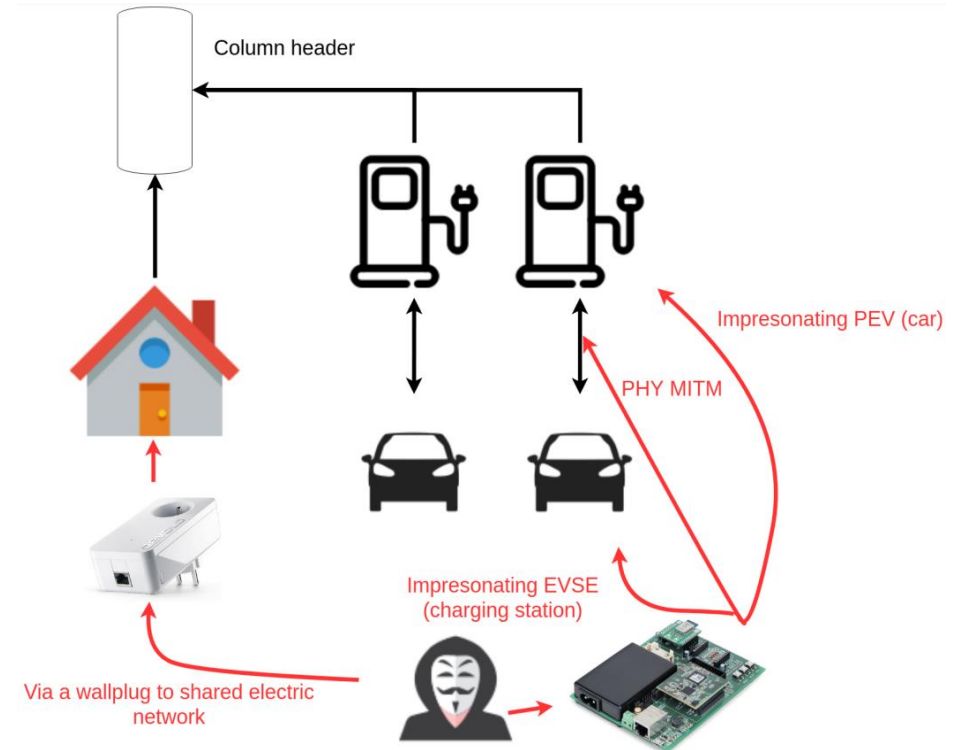
Tested solution 2019 → PKI not enforced!

But in 2023 → one client in EU got it working locally 😊!

Going further

V2G flaws

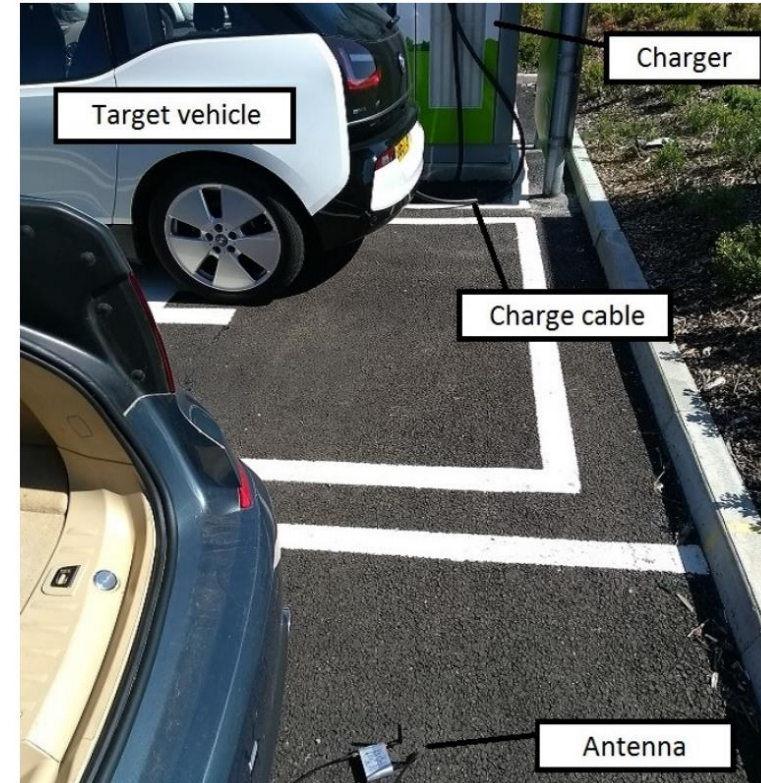
- Uses HPGP → vulnerable to key collection on powerline
- Security mode not enforced by default → MITM and injection possible
- Tools:
 - V2G Injector: <https://github.com/FIUxluS/V2GInjector>
 - HomePlugPWN: <https://github.com/FIUxluS/HomePlugPWN>



Going further

V2G key collection in radio

- HomePlug AV: hard to get the whole bandwidth with a cheap device
- But HomePlug GreenPHY as less data rate → possible with bladeRF :)



Awesome research!: <https://www.usenix.org/system/files/sec19-baker.pdf>

Conclusion

To conclude

- Vehicles embed more and more technologies
- Some of these technologies are using RF to communicate → less cables
- RF is getting more accessible to attackers
- But without proper security mechanisms:
 - Inject message to trigger bugs or fake alerts
 - Track users
 - Inject frames on CAN → needs to bypass associated gateways



Thank You

Please contact us:

 contact@penthertz.com

 +33 1 73 13 82 77

 penthertz.com

Watch us on

