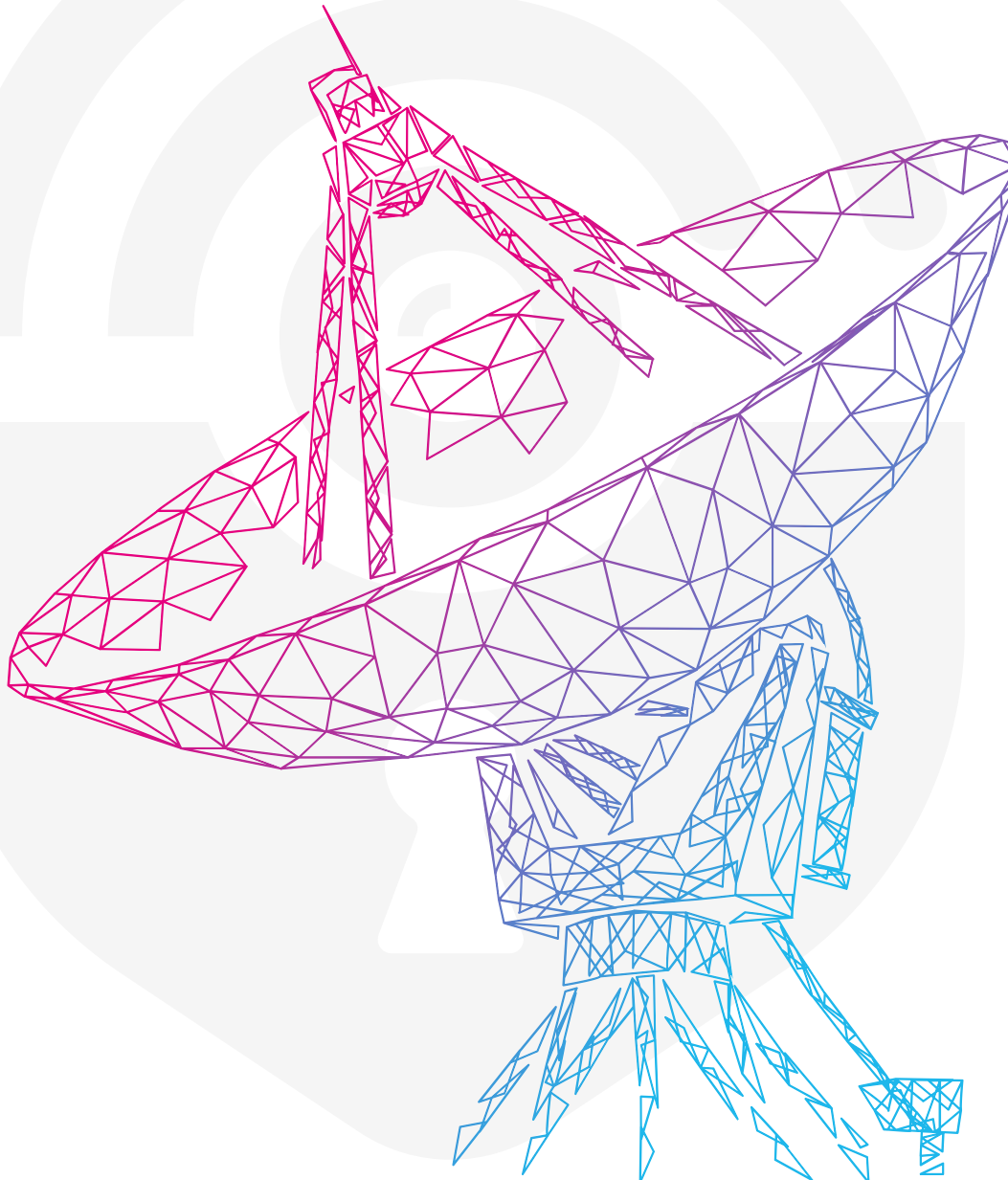
penthertz

Automotive scary signals and possible RCEs

By Sébastien Dudek

DEFCON Paris – November 2023



Founder of Penthertz

- Sébastien Dudek (@FlUxIuS)
- 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

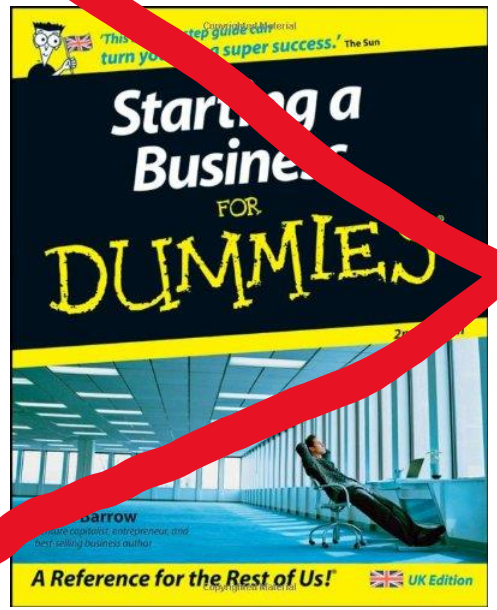
Perfect mix to make Penthertz!

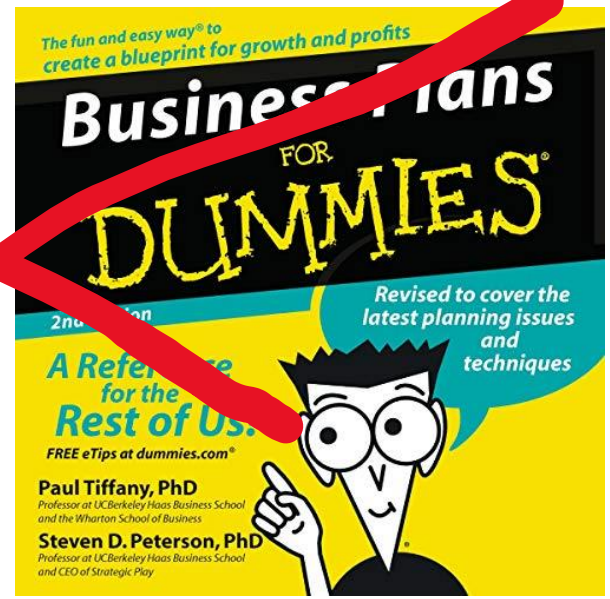
- Started the company during COVID → thinking about writing a book

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

My next book (or not)

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



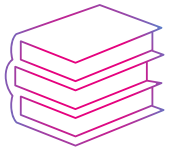


*but people have seen worse in restaurants... ☹

Main activities

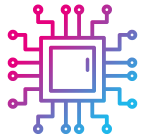
Security assessments

- 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

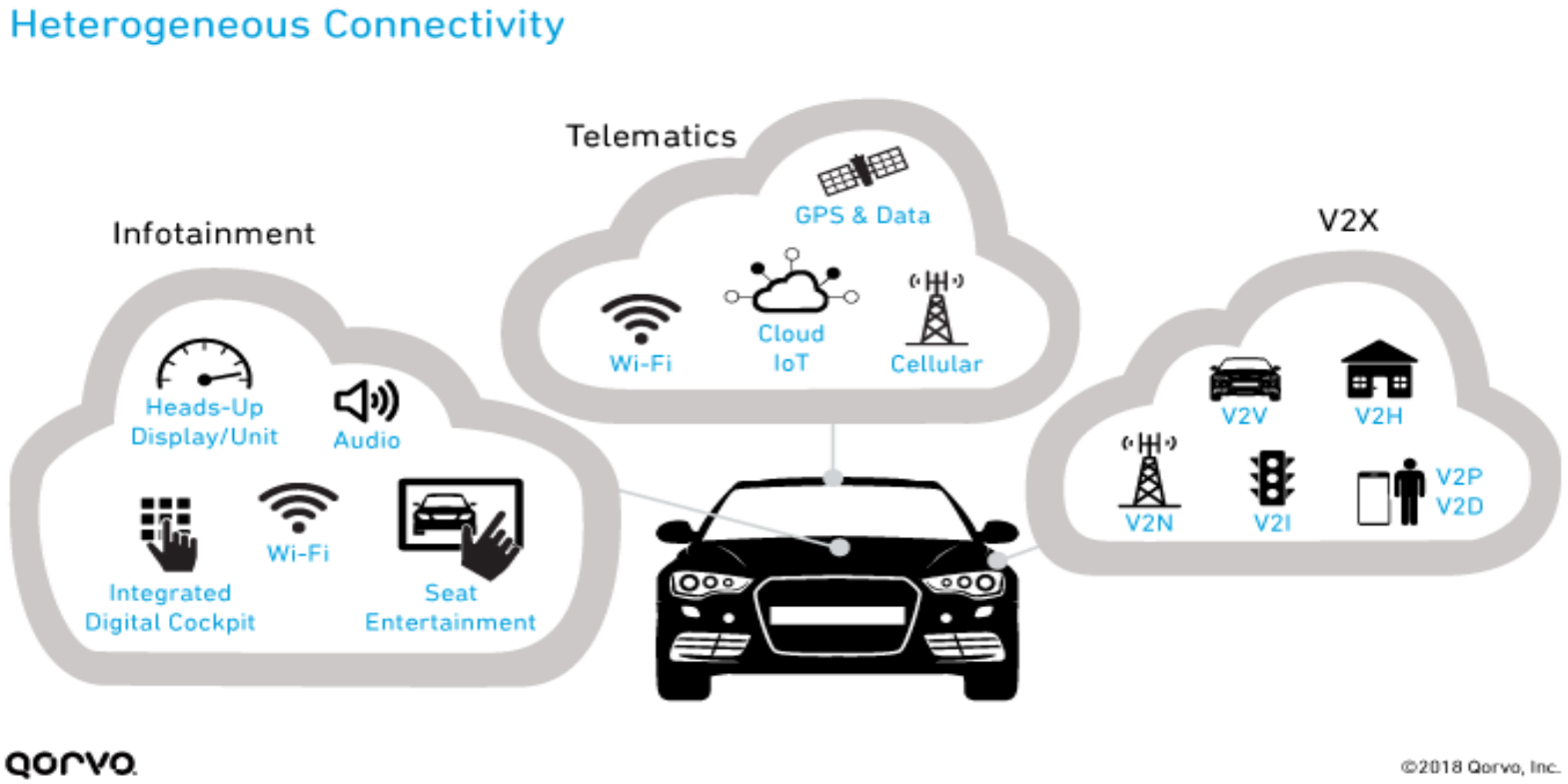
- Need to manage any type of transmission (2G-5G, Wi-Fi, Remotes, Bluetooth, ZigBee, RFID, **exotic communications**, etc.).
- Today's challenges: handling from DC to 6 or even 8 GHz with a decent stability
- Next challenges → 30 GHz at least with mmWave bands
- Able to get large bandwidth in some situation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

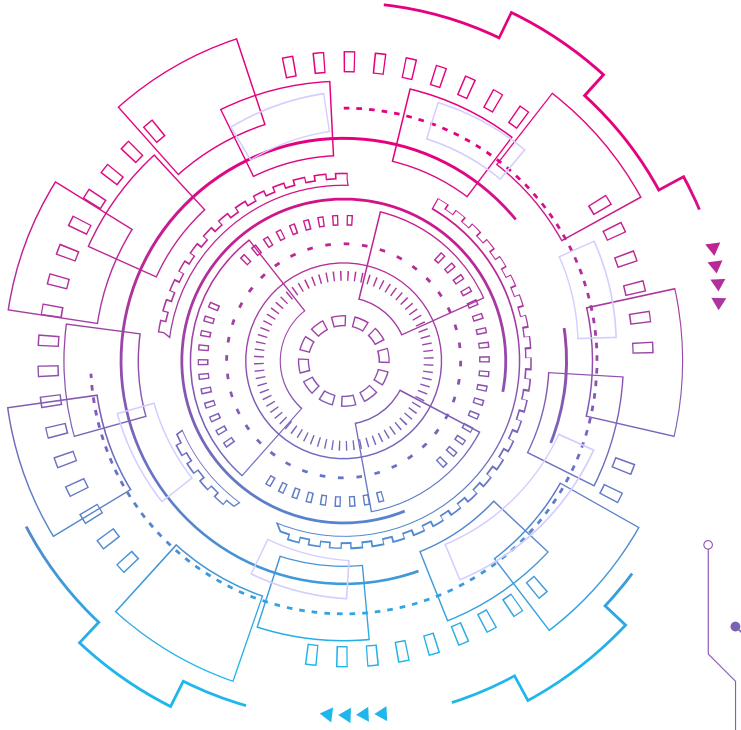
2021 Picture → the tables have never been so clean!

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

Connected cars

RF communications in cars





Summary

01 Common vectors & attacks

02 Nice opportunities to debug OTA and get root

04 Going further

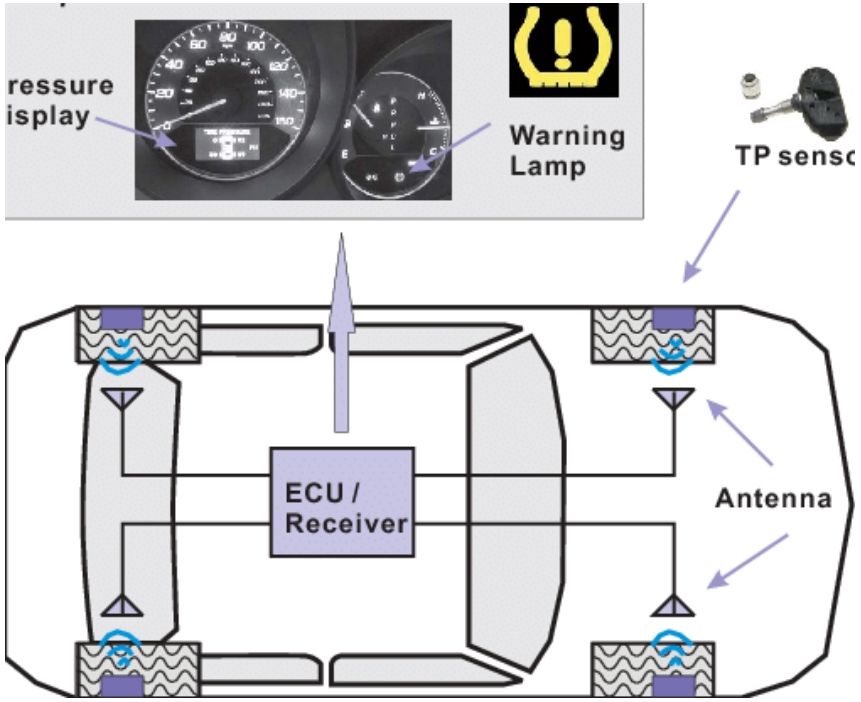
05 Conclusion

Vectors & Attacks

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

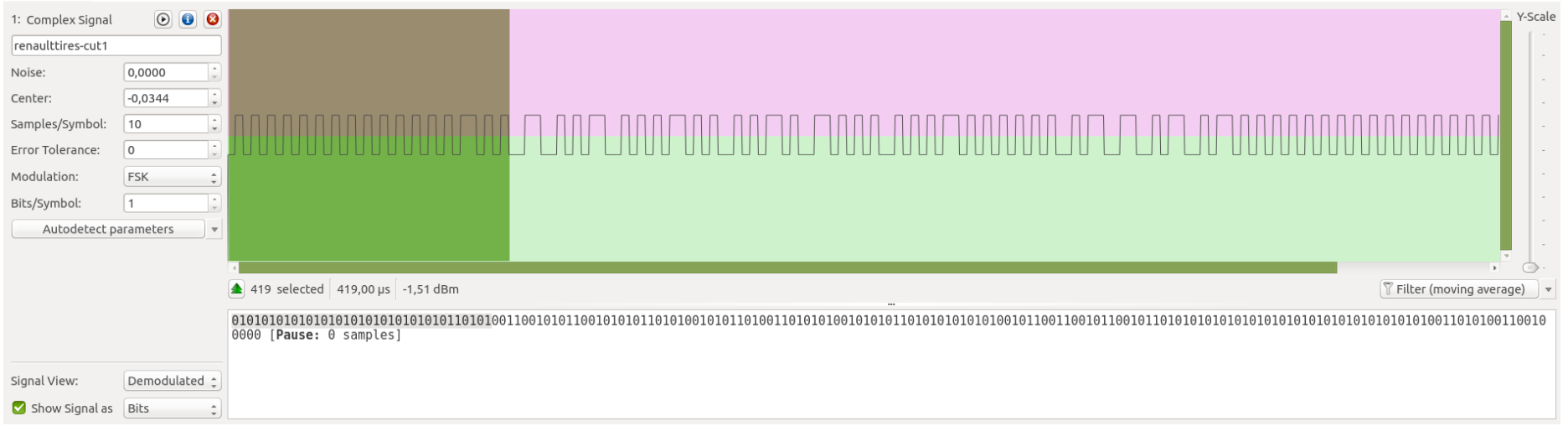
Capture

- 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)

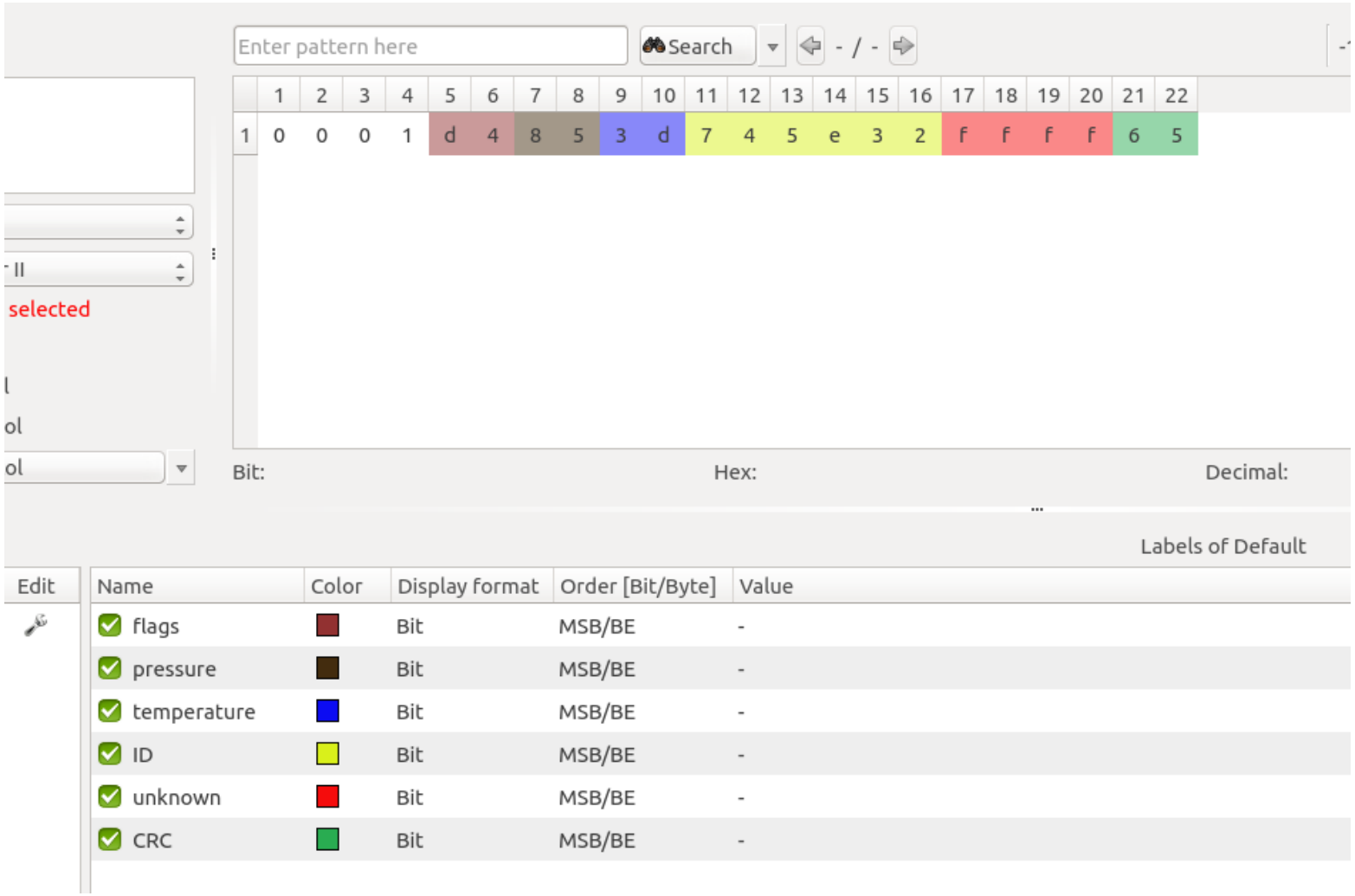
*Dedicated RF chips can be used instead of SDR = cheaper (~10€)

Demodulating data

- Quick way with URH:



Decoding the data

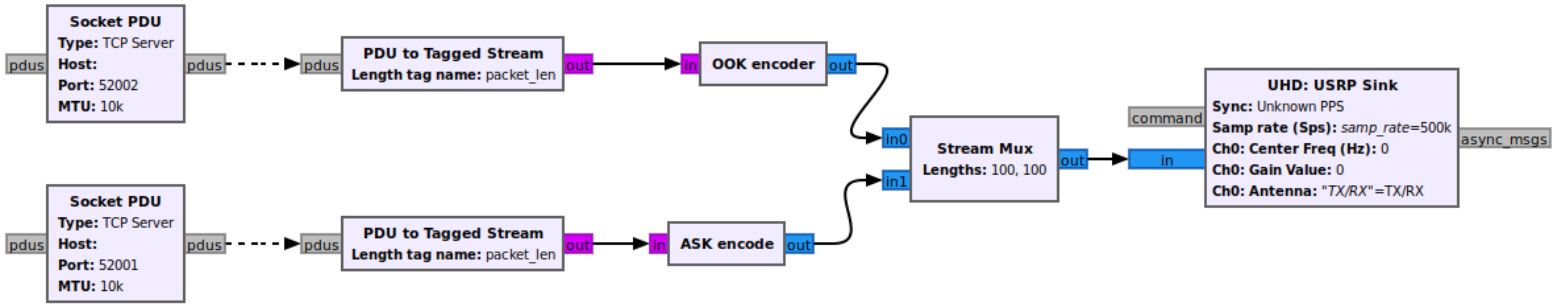


Transmission

- Requires a transmitter
- Raspberry Pi seems a cheap solution ~= 50€
- RPiTX allows transmitting over 5 KHz – 1.5 GHz

Handling two modulations

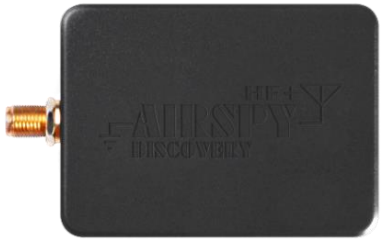
- Handling two modulations
- Hybrid sensors = more support?
- If only sending 1 modulated signal, we can also mix everything:



RF activation - captures

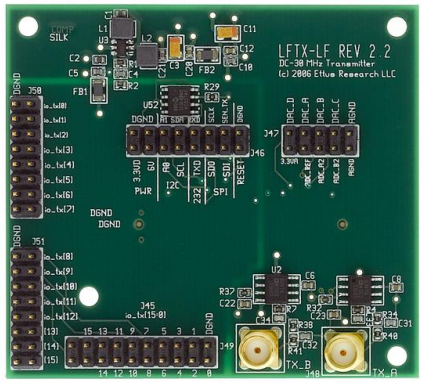
- A Low Frequency (LF) signal is sent → wake-up the sensor
- Frequency used: 125 KHz
- To receive it with RTL-SDR → need an up-converter (+50-100€)

- 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?

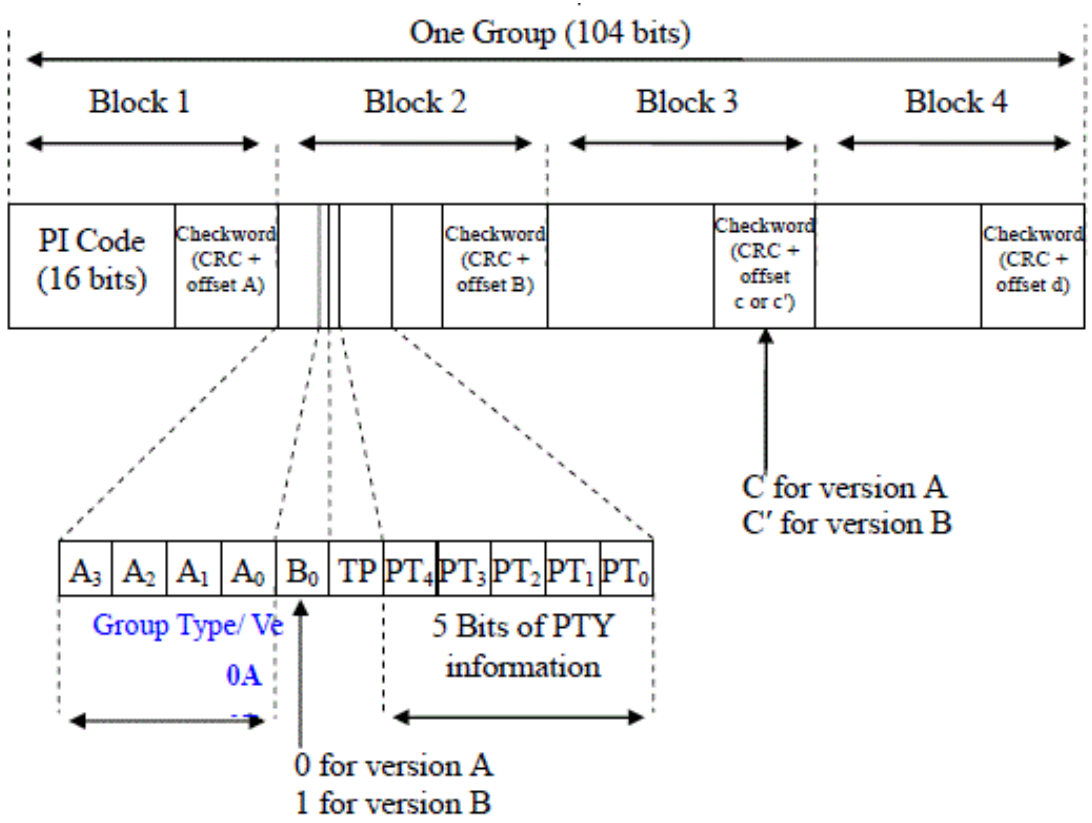
- AM and FM are just raw analogic signals → no structures
- But digital information carry:
  - Object types
  - Sometimes length of objects
  - Strings
  - ID
- There is maybe an area for fuzzing! ;)

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.

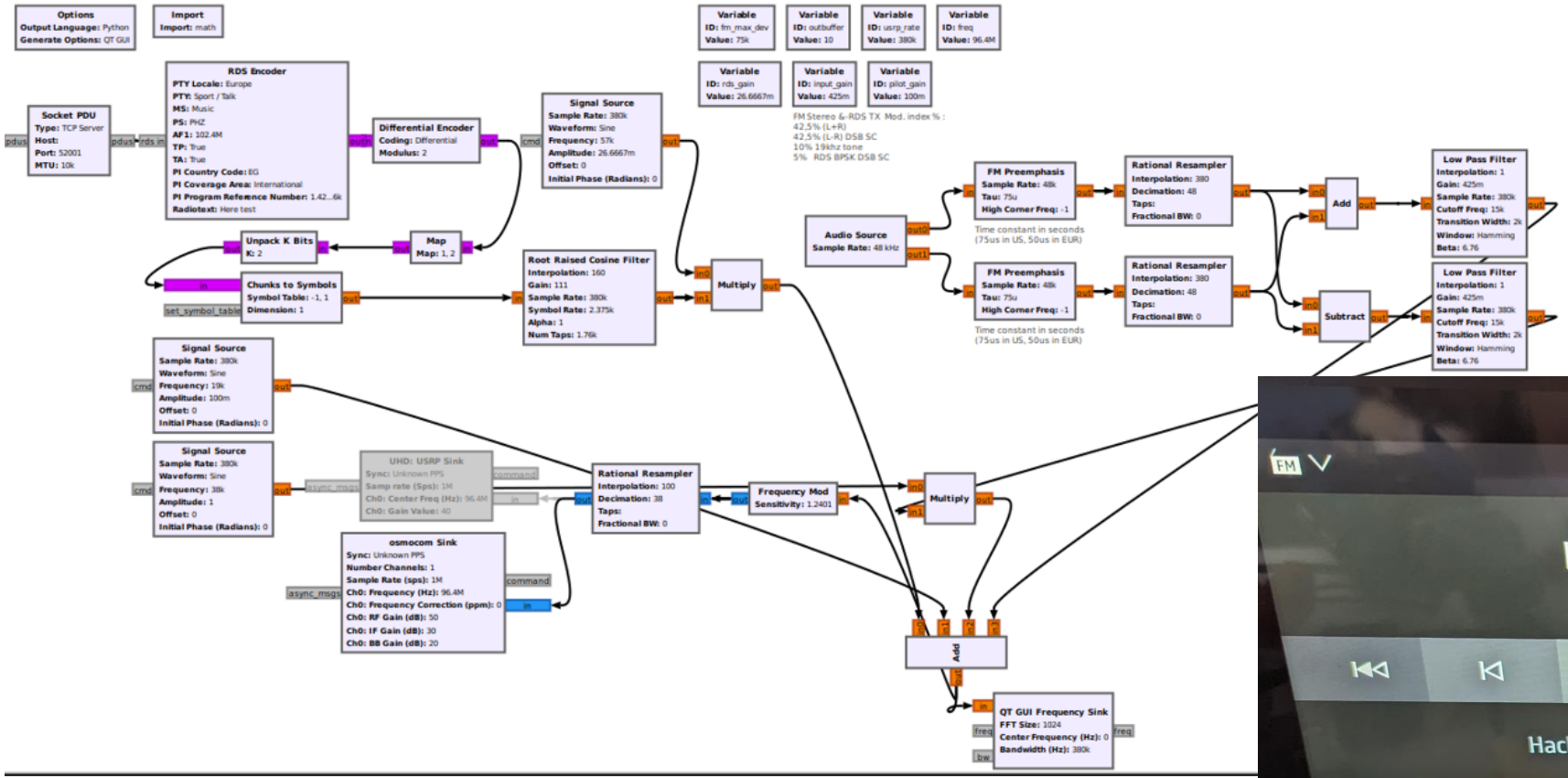


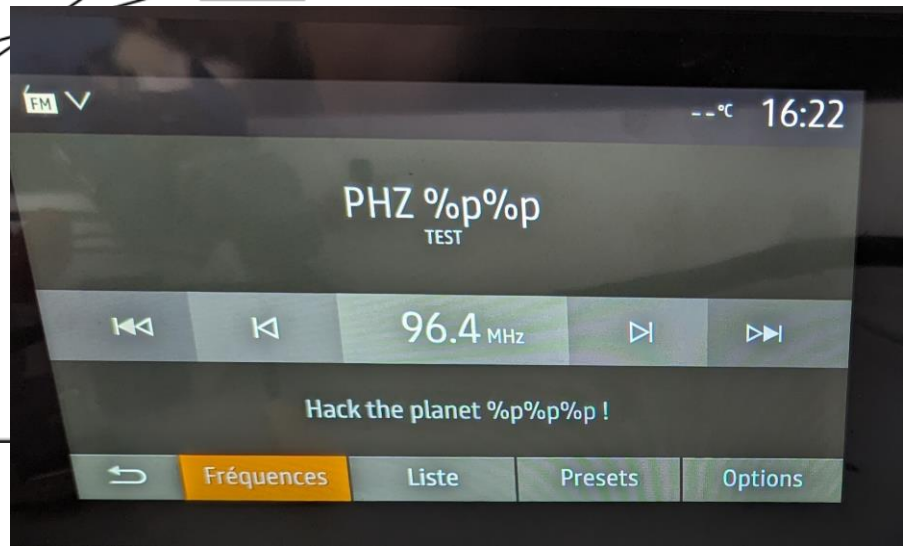
Go further by Friedt Jean-Michel: https://connect.ed-diamond.com/GNU-Linux-Magazine/glmf-204/radio-data-system-rds-analyse-du-canal-numerique-transmis-par-les-stations-radio-fm-commerciales-introduction-aux-codes-correcteurs-d-erreur

DAB

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

RDS injections attempts

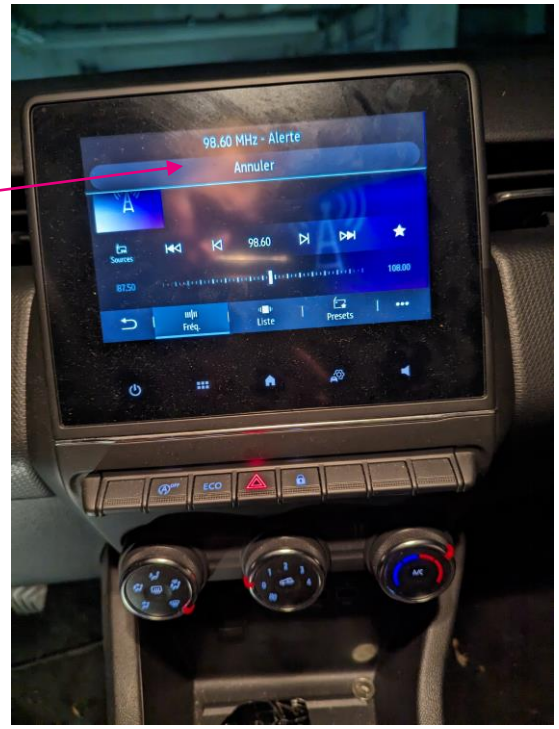




*with a modified gr-rds + libs ;)

RDS Alerts

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



RDS Alerts

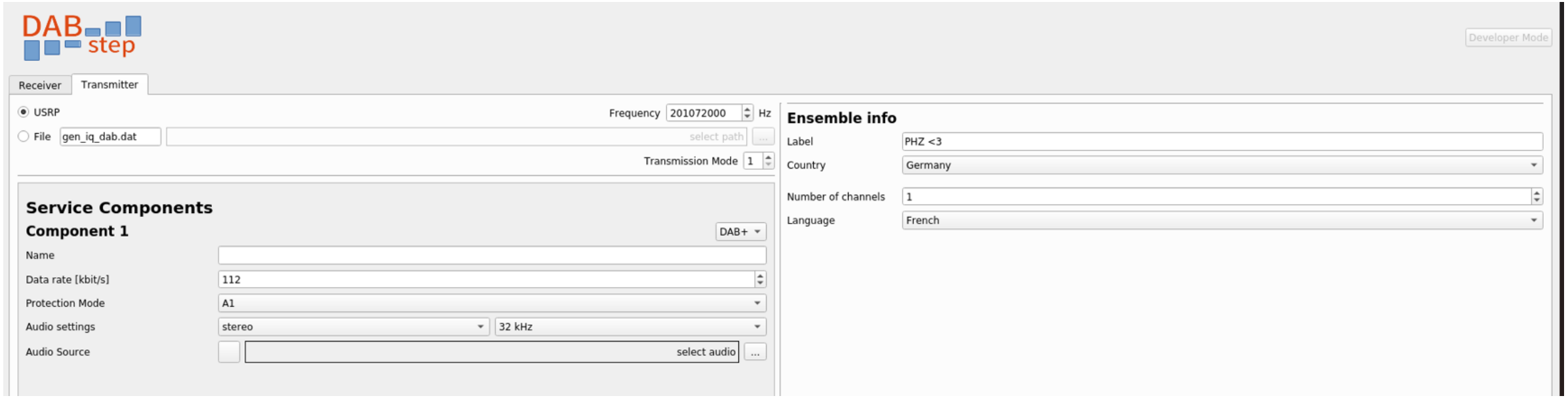
- Many events exist:

```
{"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

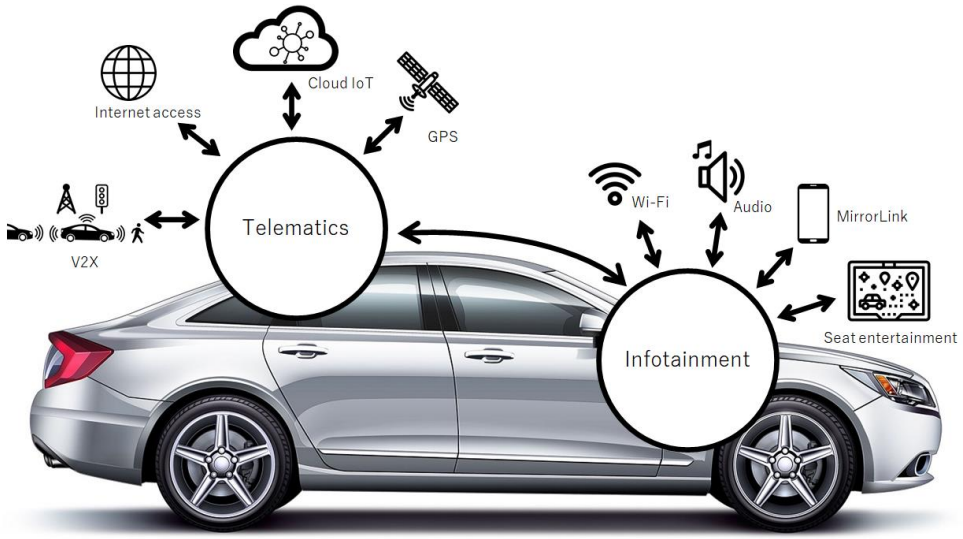
- Same with a modified “DAB step”! :



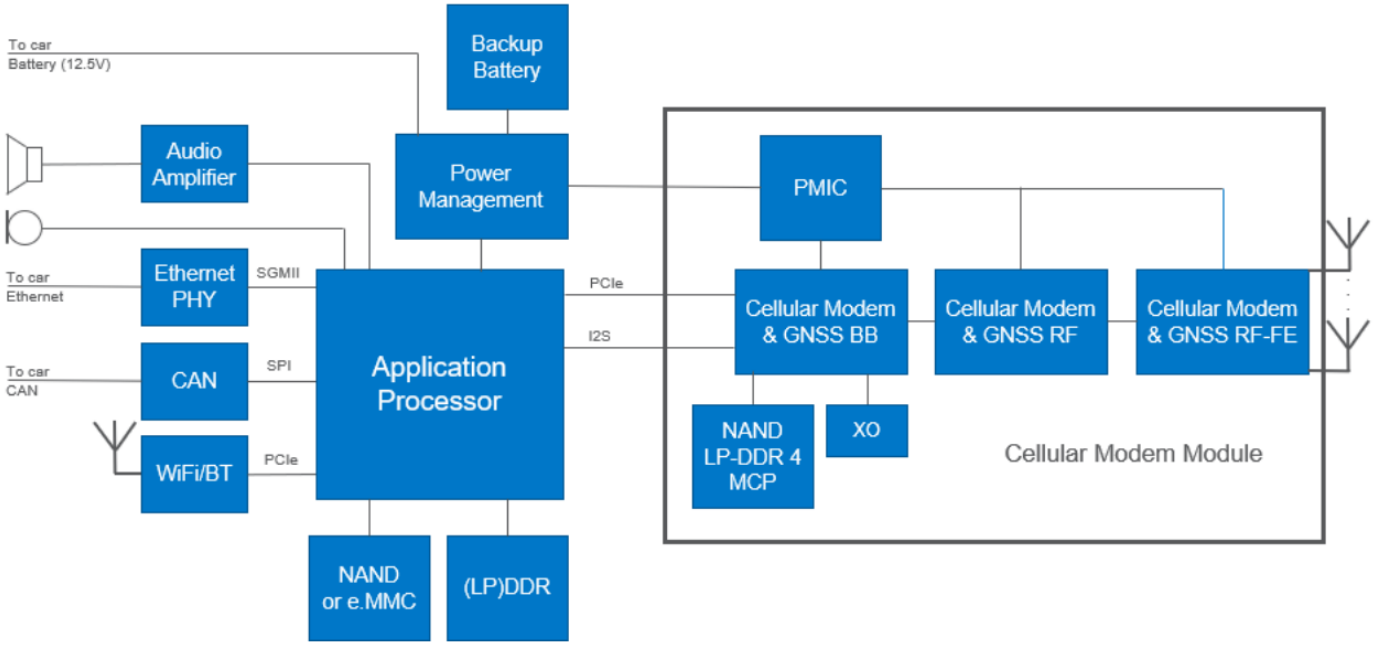
Mobile access on IVI

TCUs with 3G-5G stacks used in cars

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



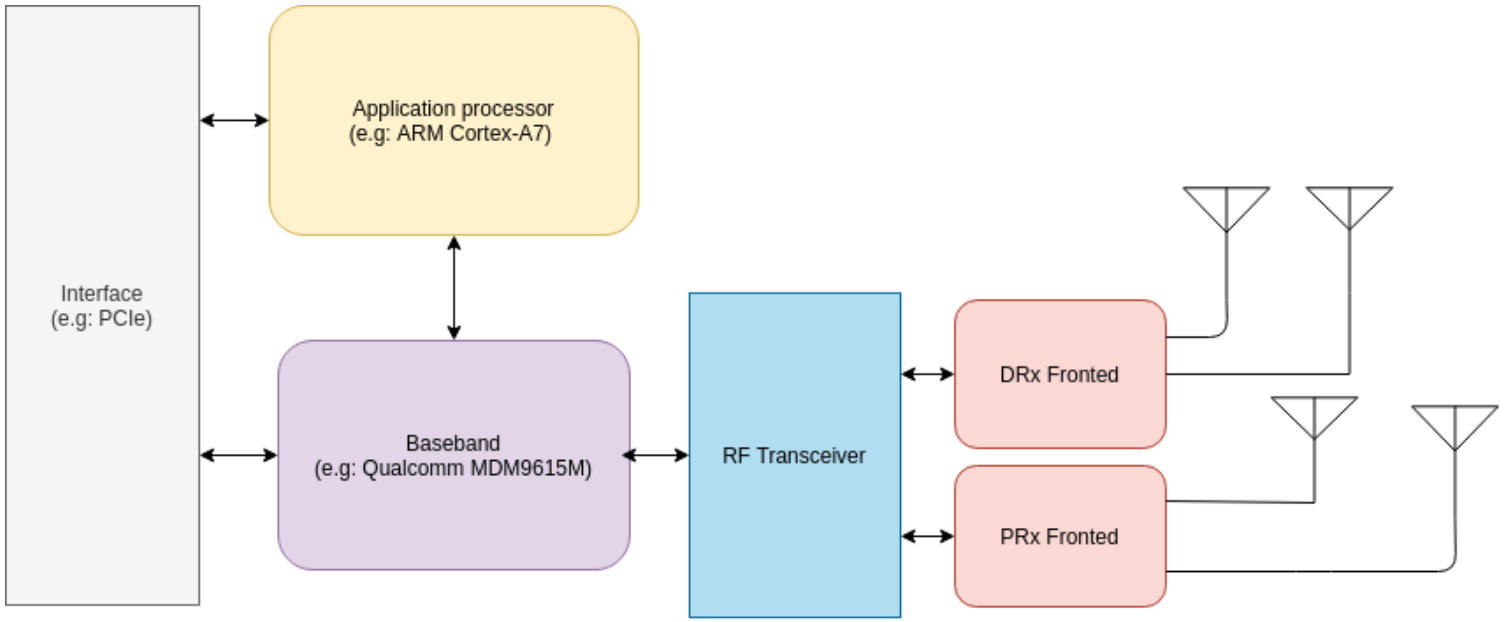
Source: https://www.i-pex.com/



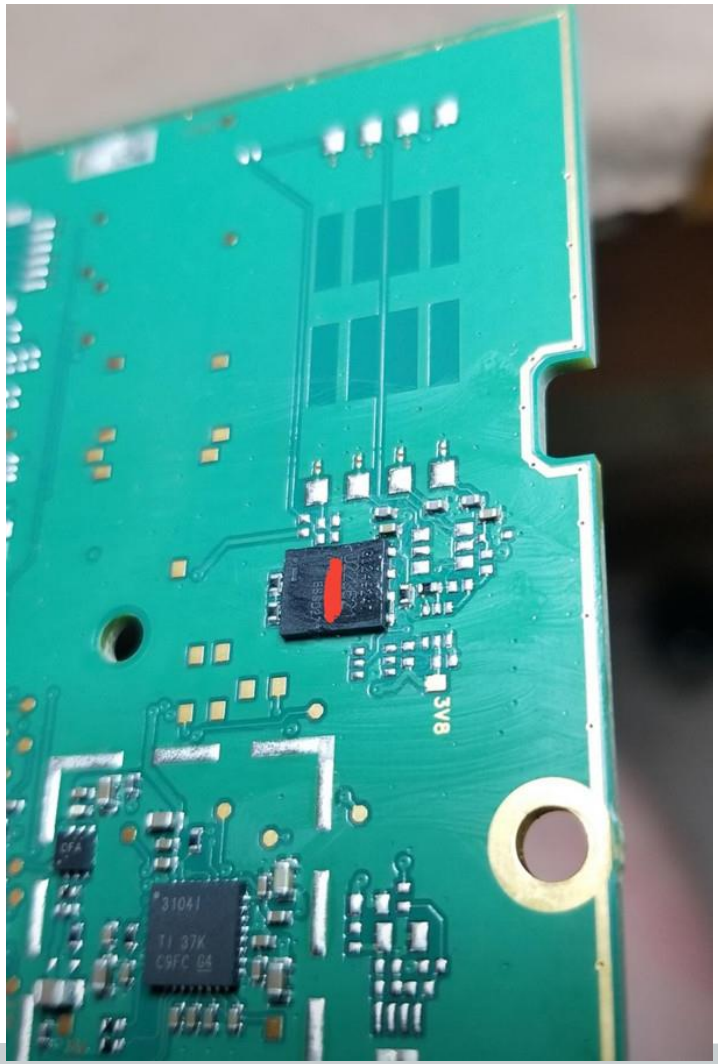
Source: https://media-www.micron.com/

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…



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	1.8/3.0V voltage domain, all (U)SIM interfaces should be protected against ESD. If unused, please keep open
(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	
(U)SIM2_PWR	48	PO	Power supply for (U)SIM2 card	
(U)SIM2_DATA	42	DIO	(U)SIM2 card data, which has been pulled up to (U)SIM2_VDD via a 20KR resistor internally	
(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)

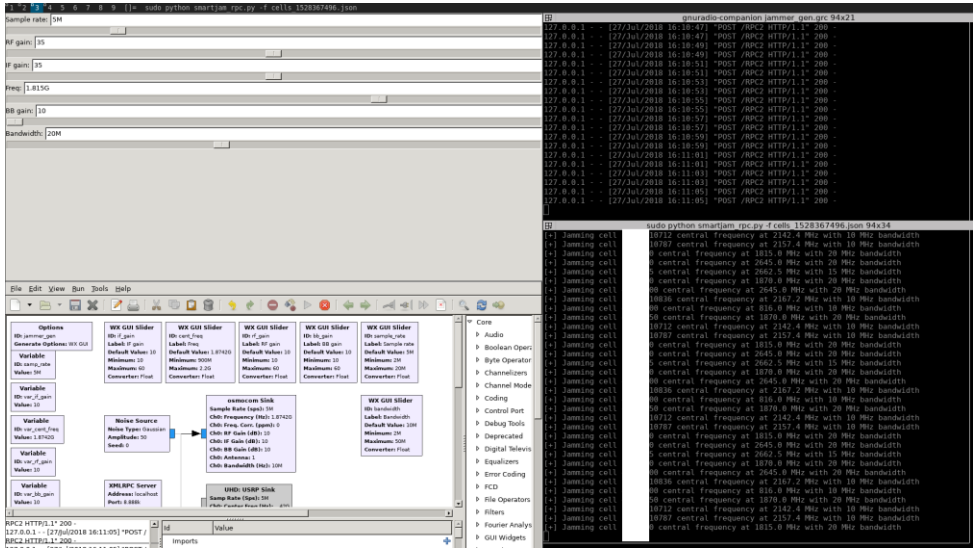
Possible attacks

- 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

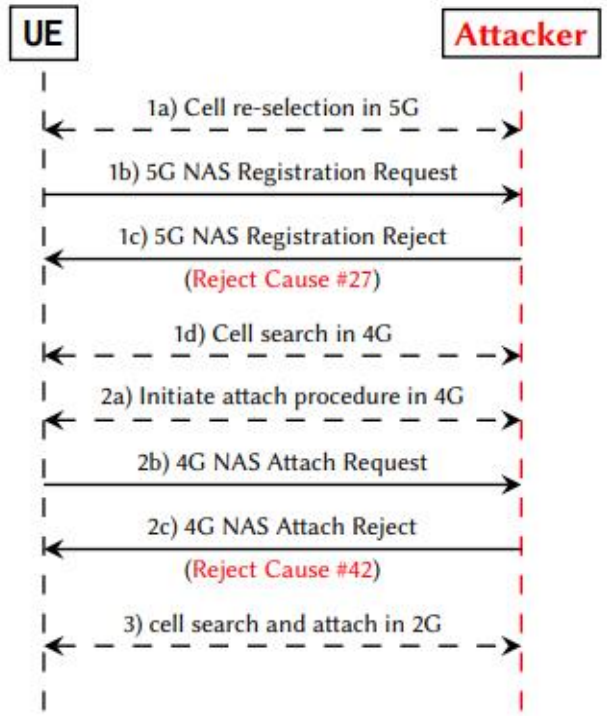
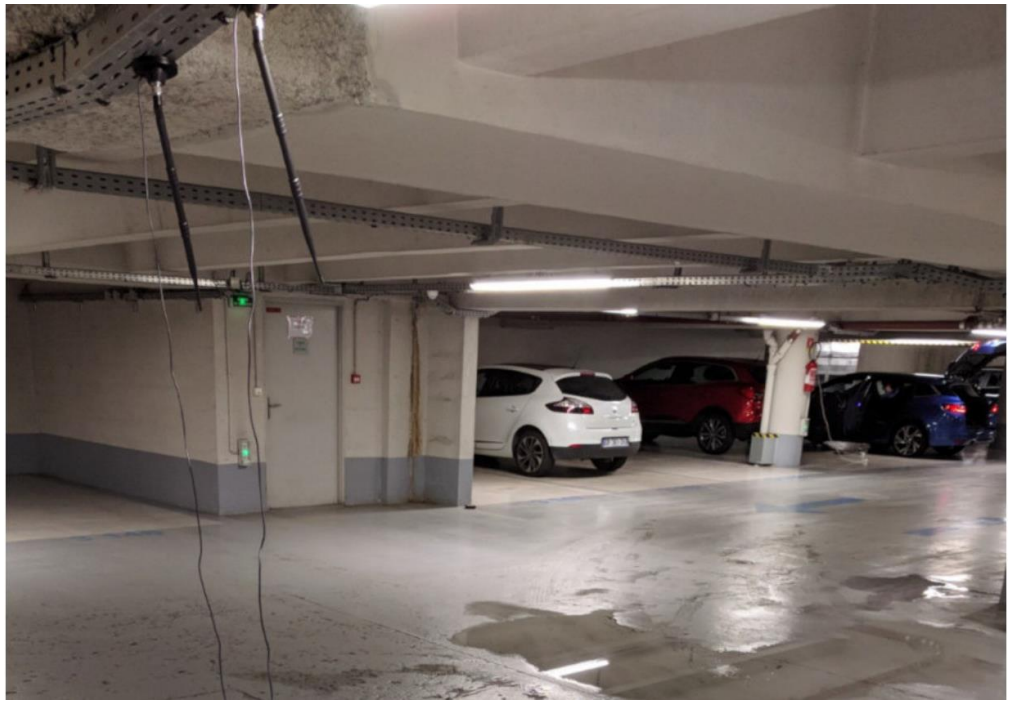


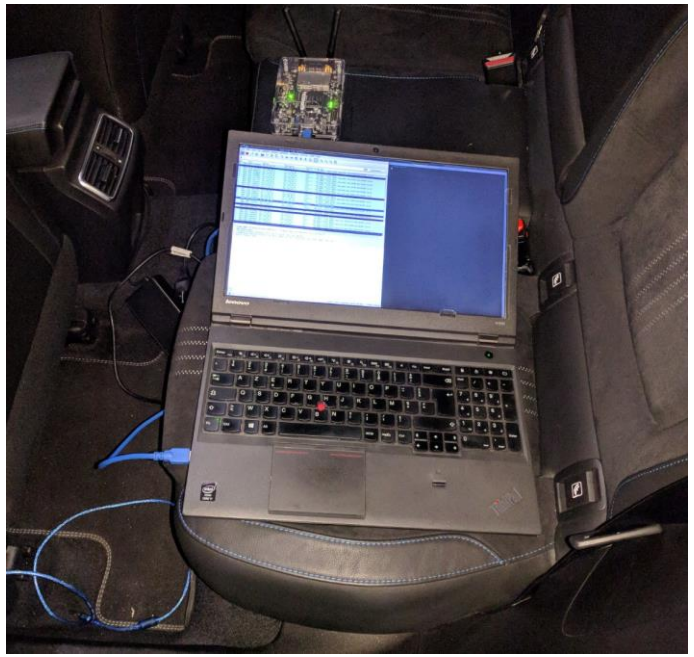
Figure 3: Protocol flow of downgrade dance from 5G to 2G.

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	991.484923306	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

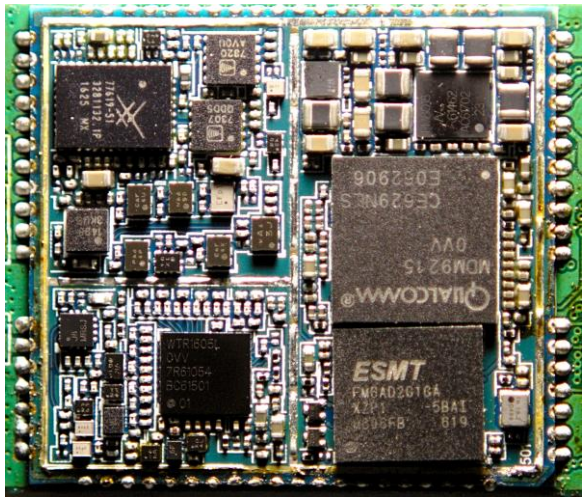
Going further on the backend

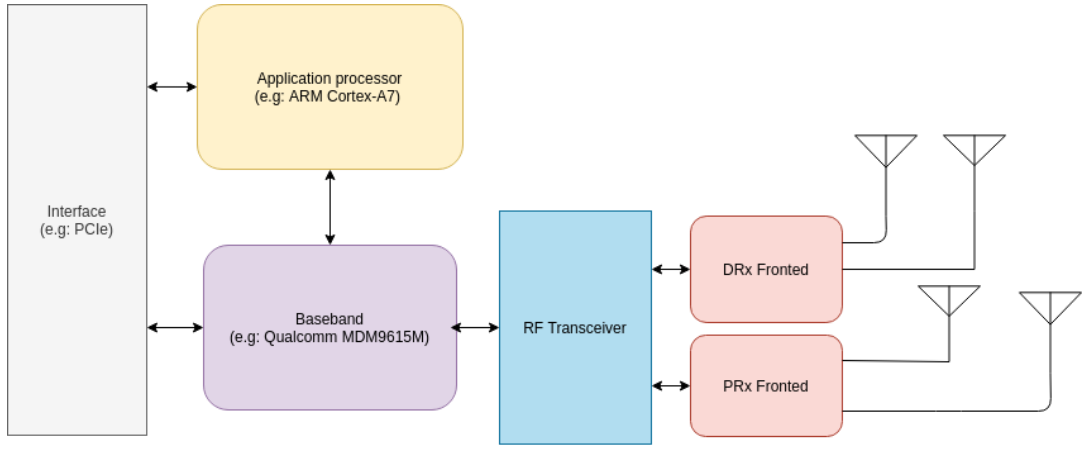
- We can try attacking the backend
- We can extract the eSIM and readapt pins for a modem:

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

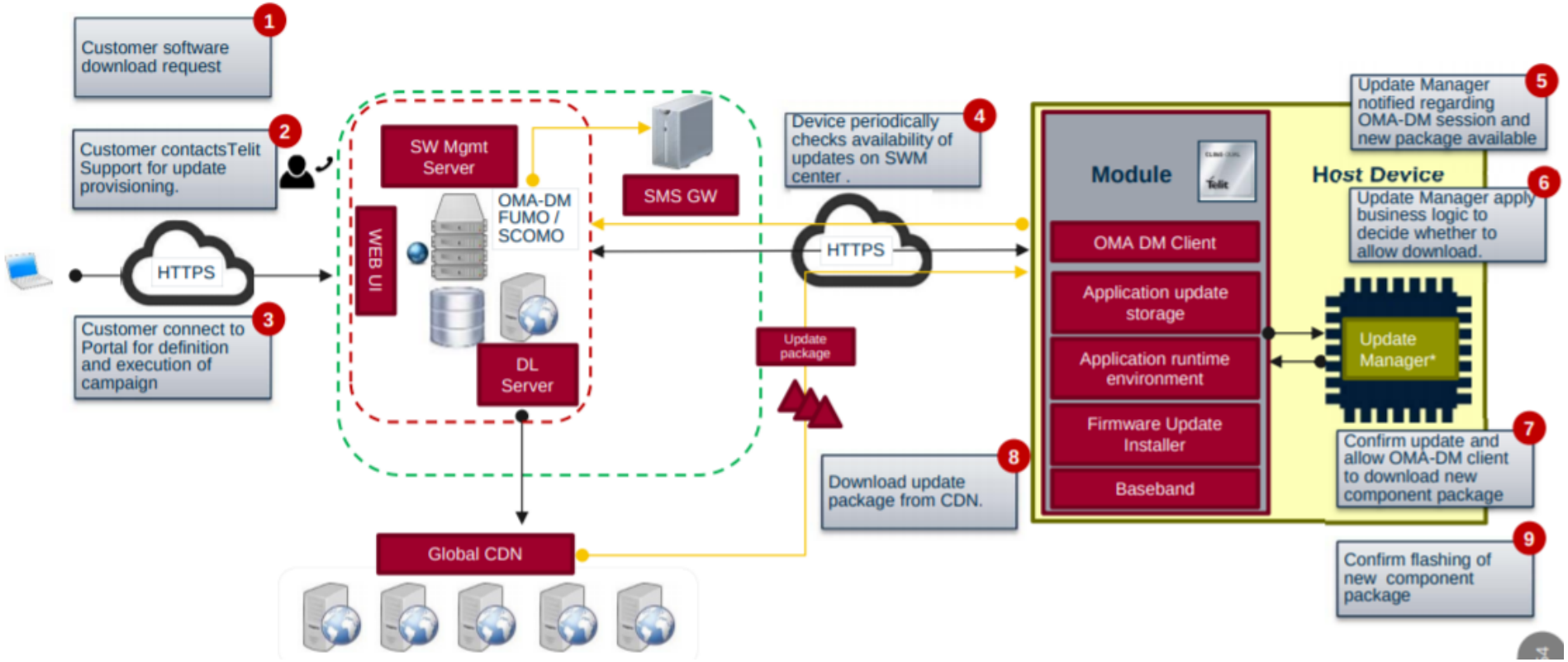
Mobile modules

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

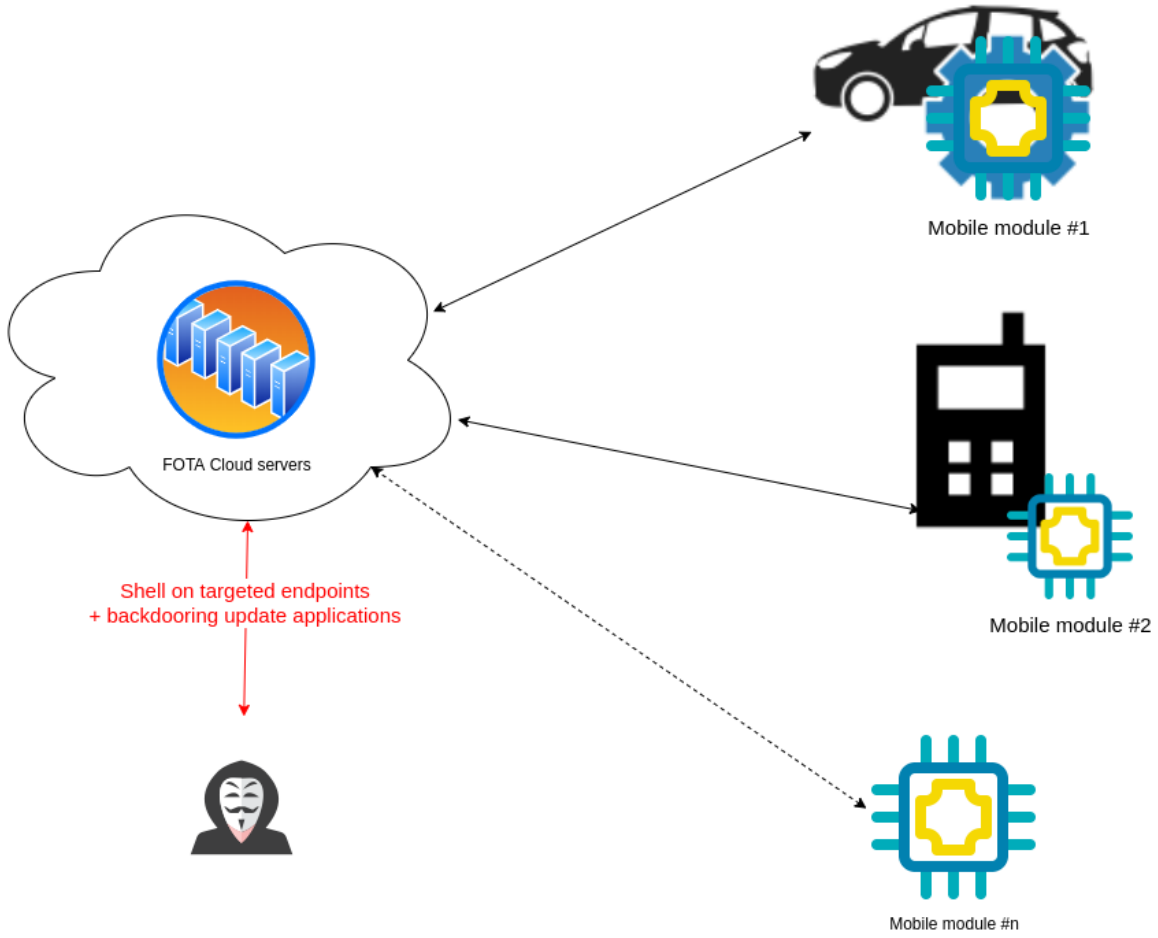




FOTA updates: schema



FOTA updates: Impacts



Some nice opportunities

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

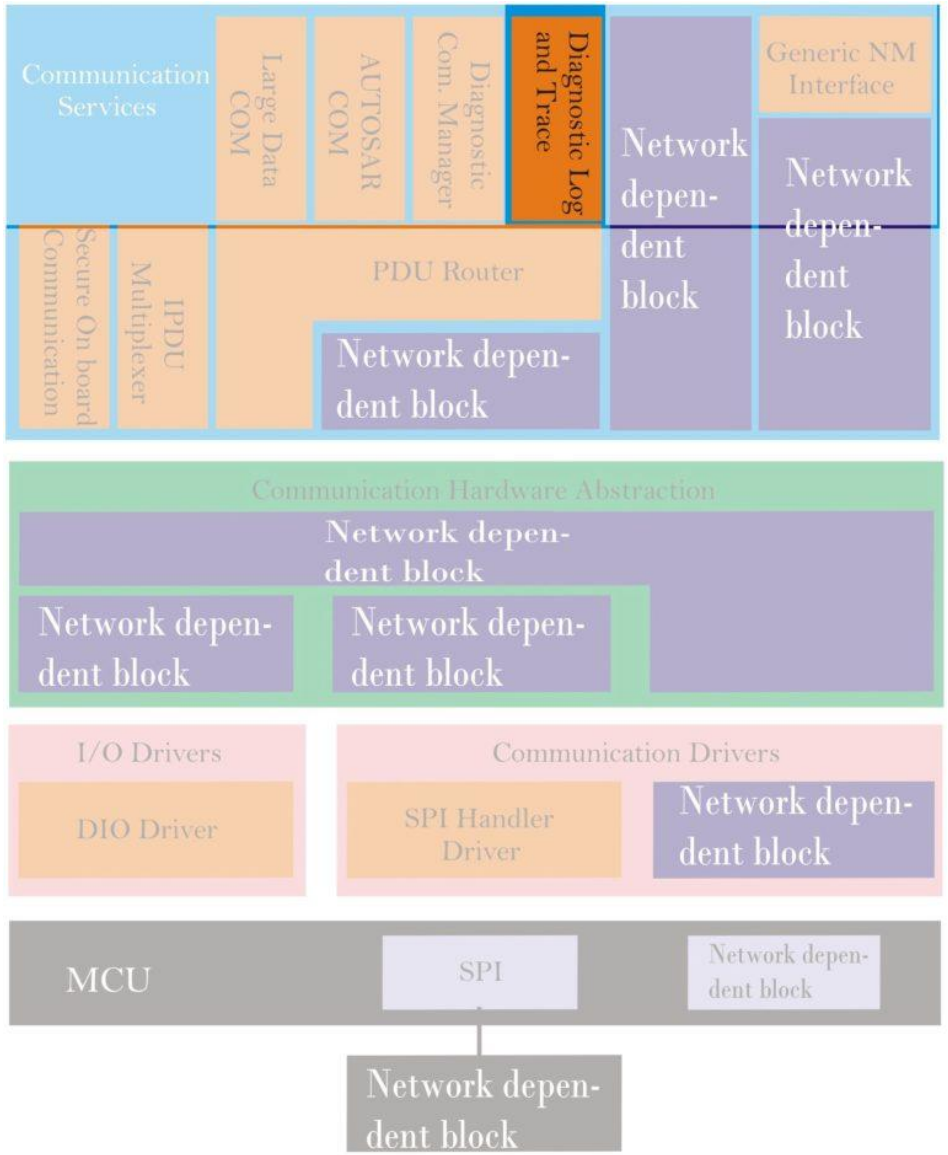
- QNX in uses:
  - Look at exposed **qconn** service ☺ (good old trick! But with a little update)

```
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?

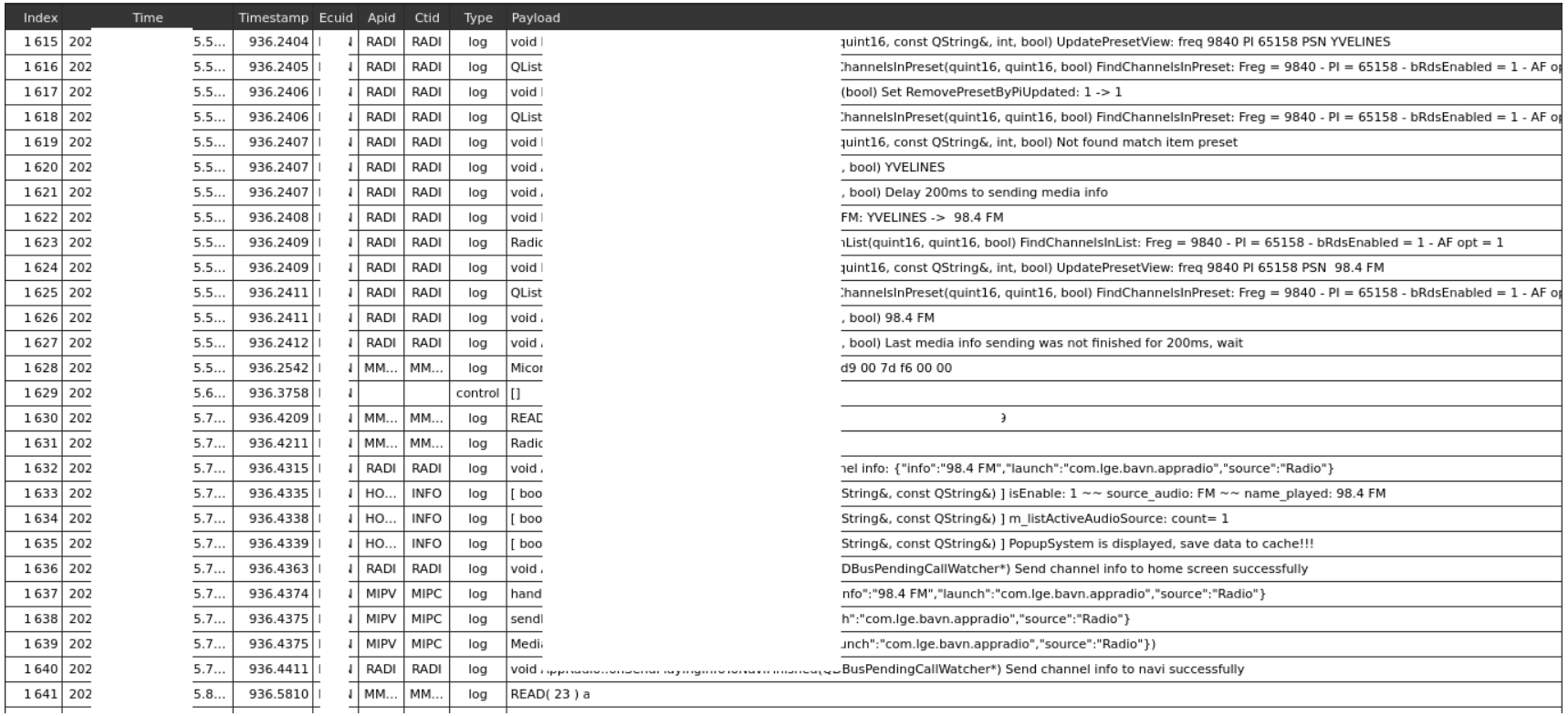
- Diagnostic Log and Trace
- Sender-receiver communication
- See more: https://autosartutorials.com/diagnostic-log-and-trace/



Some nice opportunities

DLT traces

- Can trace:
  - Events
  - Crashes
  - Running processes



Index	Time	Timestamp	Ecuid	Apid	Ctid	Type	Payload
1 615	202 5.5...	936.2404	I	RADI	RADI	log	void quint16, const QString&, int, bool) UpdatePresetView: freq 9840 PI 65158 PSN YVELINES
1 616	202 5.5...	936.2405	I	RADI	RADI	log	QList hannelsInPreset(quint16, quint16, bool) FindChannelsInPreset: Freq = 9840 - PI = 65158 - bRdsEnabled = 1 - AF op
1 617	202 5.5...	936.2406	I	RADI	RADI	log	void (bool) Set RemovePresetByPiUpdated: 1 -> 1
1 618	202 5.5...	936.2406	I	RADI	RADI	log	QList hannelsInPreset(quint16, quint16, bool) FindChannelsInPreset: Freq = 9840 - PI = 65158 - bRdsEnabled = 1 - AF op
1 619	202 5.5...	936.2407	I	RADI	RADI	log	void quint16, const QString&, int, bool) Not found match item preset
1 620	202 5.5...	936.2407	I	RADI	RADI	log	void , bool) YVELINES
1 621	202 5.5...	936.2407	I	RADI	RADI	log	void , bool) Delay 200ms to sending media info
1 622	202 5.5...	936.2408	I	RADI	RADI	log	void FM: YVELINES -> 98.4 FM
1 623	202 5.5...	936.2409	I	RADI	RADI	log	Radic nList(quint16, quint16, bool) FindChannelsInList: Freq = 9840 - PI = 65158 - bRdsEnabled = 1 - AF opt = 1
1 624	202 5.5...	936.2409	I	RADI	RADI	log	void quint16, const QString&, int, bool) UpdatePresetView: freq 9840 PI 65158 PSN 98.4 FM
1 625	202 5.5...	936.2411	I	RADI	RADI	log	QList hannelsInPreset(quint16, quint16, bool) FindChannelsInPreset: Freq = 9840 - PI = 65158 - bRdsEnabled = 1 - AF op
1 626	202 5.5...	936.2411	I	RADI	RADI	log	void , bool) 98.4 FM
1 627	202 5.5...	936.2412	I	RADI	RADI	log	void , bool) Last media info sending was not finished for 200ms, wait
1 628	202 5.5...	936.2542	I	MM...	MM...	log	Micor d9 00 7d f6 00 00
1 629	202 5.6...	936.3758	I			control	[]
1 630	202 5.7...	936.4209	I	MM...	MM...	log	READ)
1 631	202 5.7...	936.4211	I	MM...	MM...	log	Radic
1 632	202 5.7...	936.4315	I	RADI	RADI	log	void el info: {"info":"98.4 FM","launch":"com.lge.bavn.appradio","source":"Radio"}
1 633	202 5.7...	936.4335	I	HO...	INFO	log	[boo String&, const QString&)] isEnable: 1 ~~ source_audio: FM ~~ name_played: 98.4 FM
1 634	202 5.7...	936.4338	I	HO...	INFO	log	[boo String&, const QString&)] m_listActiveAudioSource: count= 1
1 635	202 5.7...	936.4339	I	HO...	INFO	log	[boo String&, const QString&)] PopupSystem is displayed, save data to cache!!!
1 636	202 5.7...	936.4363	I	RADI	RADI	log	void DBusPendingCallWatcher*) Send channel info to home screen successfully
1 637	202 5.7...	936.4374	I	MIPV	MIPC	log	hand nfo":"98.4 FM","launch":"com.lge.bavn.appradio","source":"Radio"}
1 638	202 5.7...	936.4375	I	MIPV	MIPC	log	sendI h":"com.lge.bavn.appradio","source":"Radio"}
1 639	202 5.7...	936.4375	I	MIPV	MIPC	log	Medi unch":"com.lge.bavn.appradio","source":"Radio"})
1 640	202 5.7...	936.4411	I	RADI	RADI	log	void BusPendingCallWatcher*) Send channel info to navi successfully
1 641	202 5.8...	936.5810	I	MM...	MM...	log	READ(23) a

- Perfect to debug fuzzing when it’s exposed! ☺

Some nice opportunities

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
```

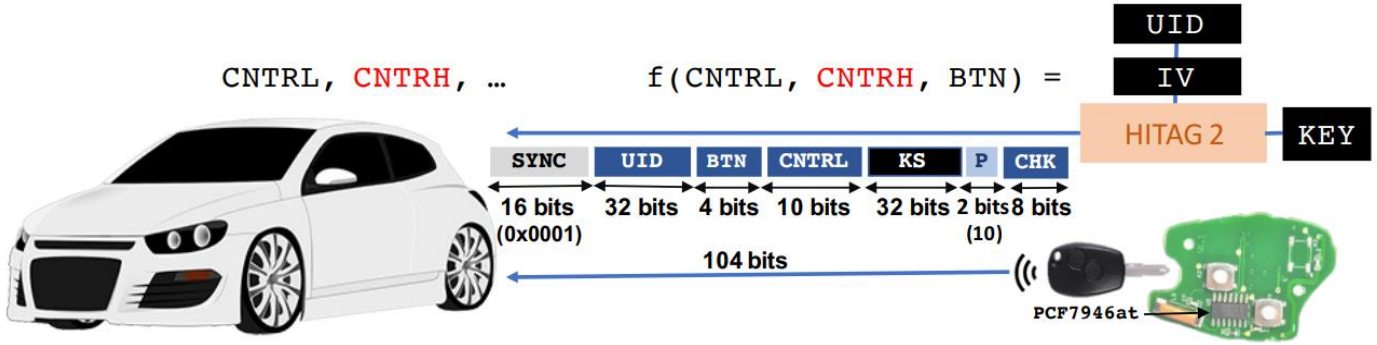
RKE systems

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

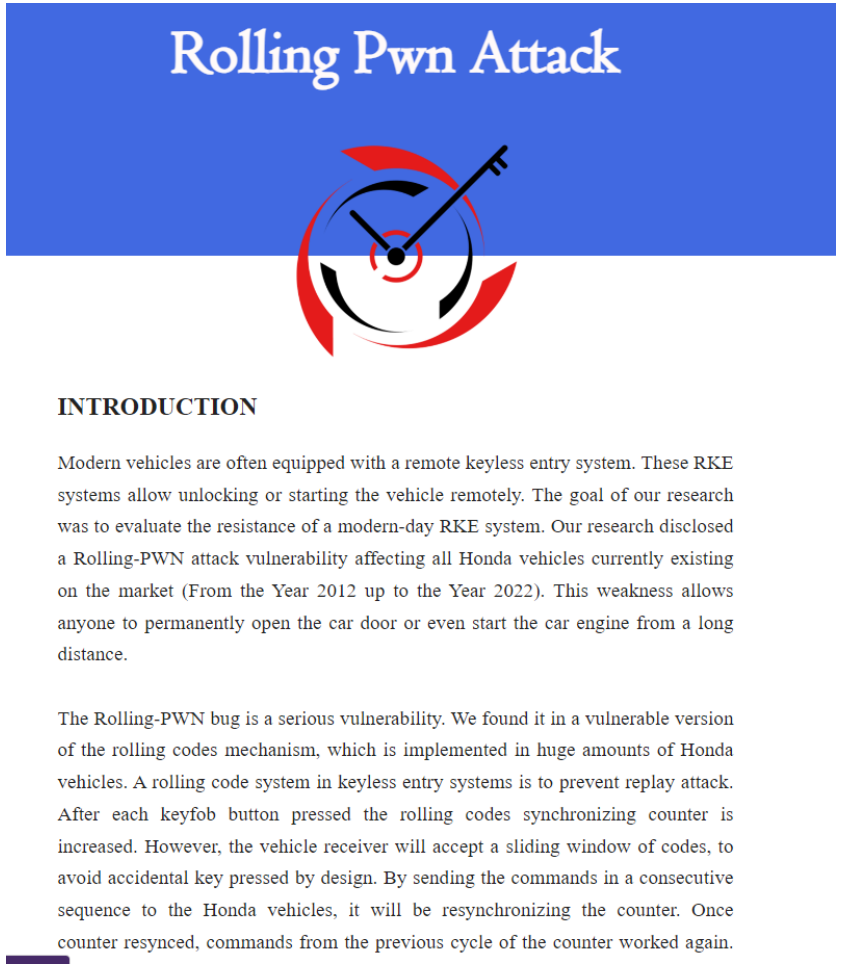


Source: From Academia to Real World : a Practical Guide to Hitag-2 RKE System Analysis by Ryad Benadjila, Mathieu Renard, José Lopes-Esteves, Chaouki Kasmi

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

RKE vs Rollback attacks

- Sometimes replaying old consecutive code → roll back



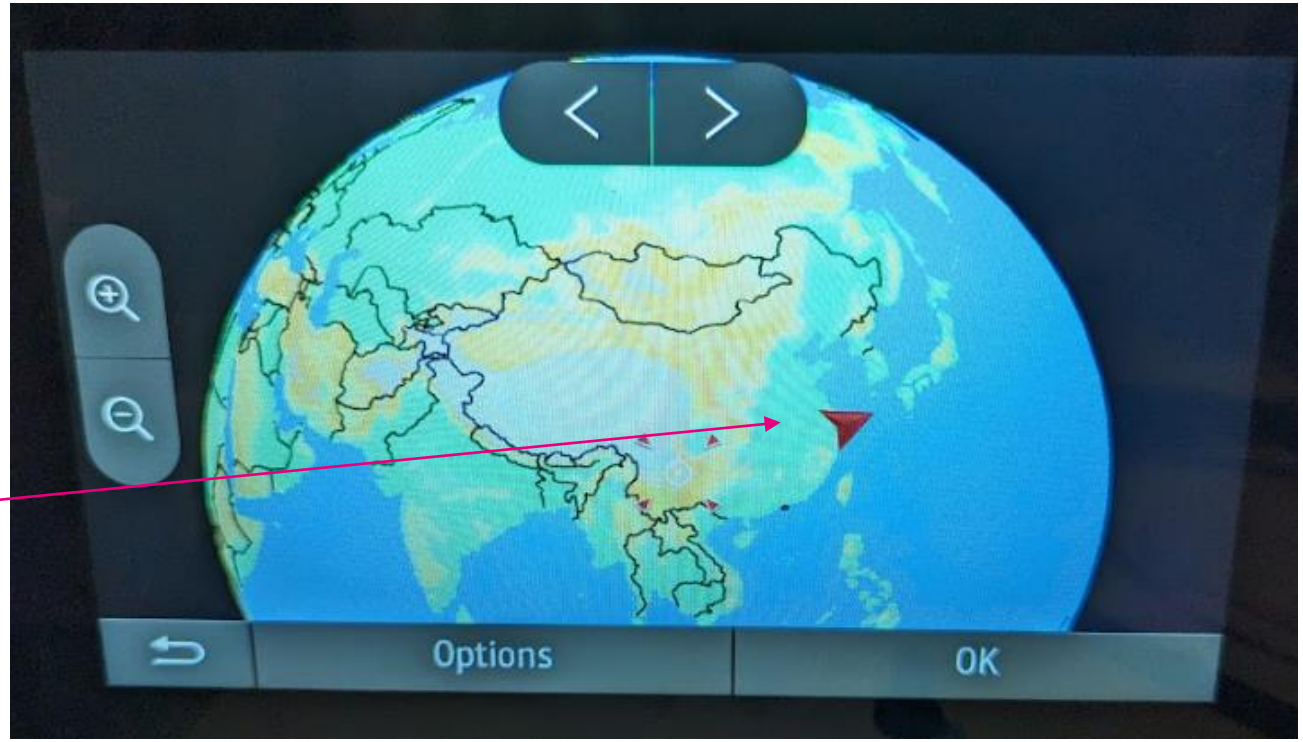
Car apps

- 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!



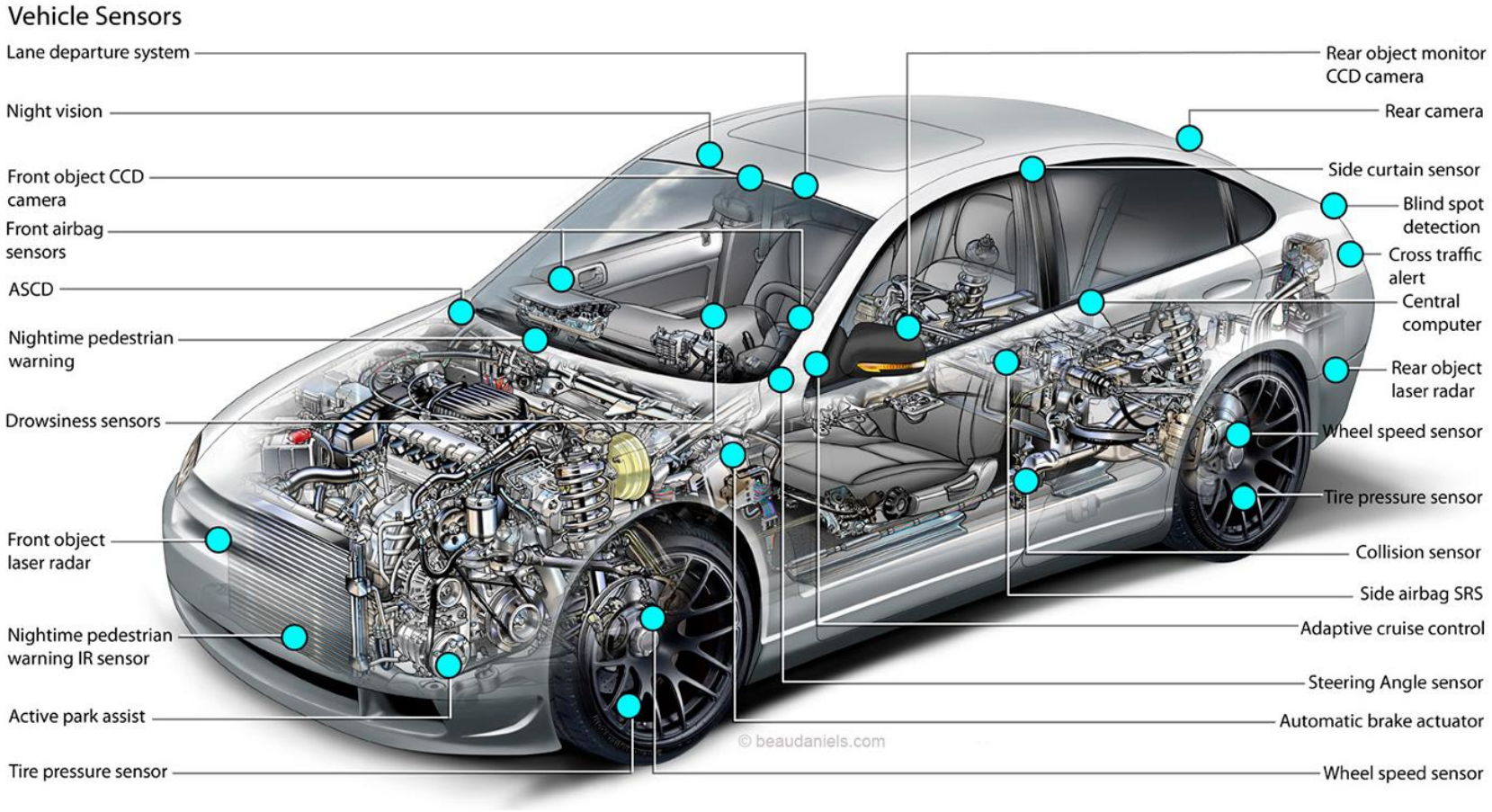
GPS against autopilot?

Hijacking vs Autopilot

- Question: What about Autopilot?

Going further

Sensors in cars > ~200

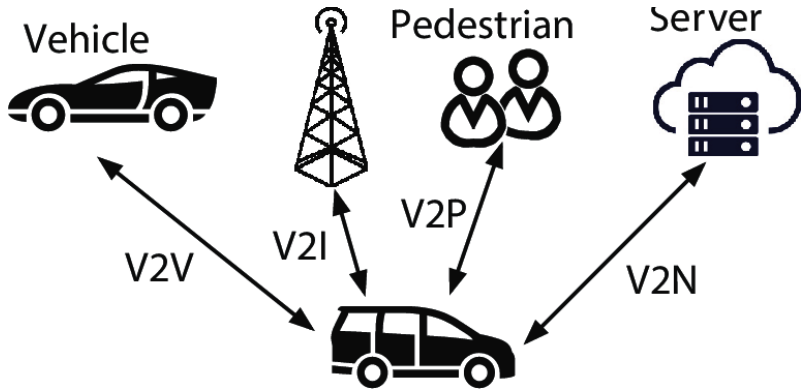


Vehicle Sensors, the connected car - https://www.behance.net/gallery/51718817/Connected-car

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

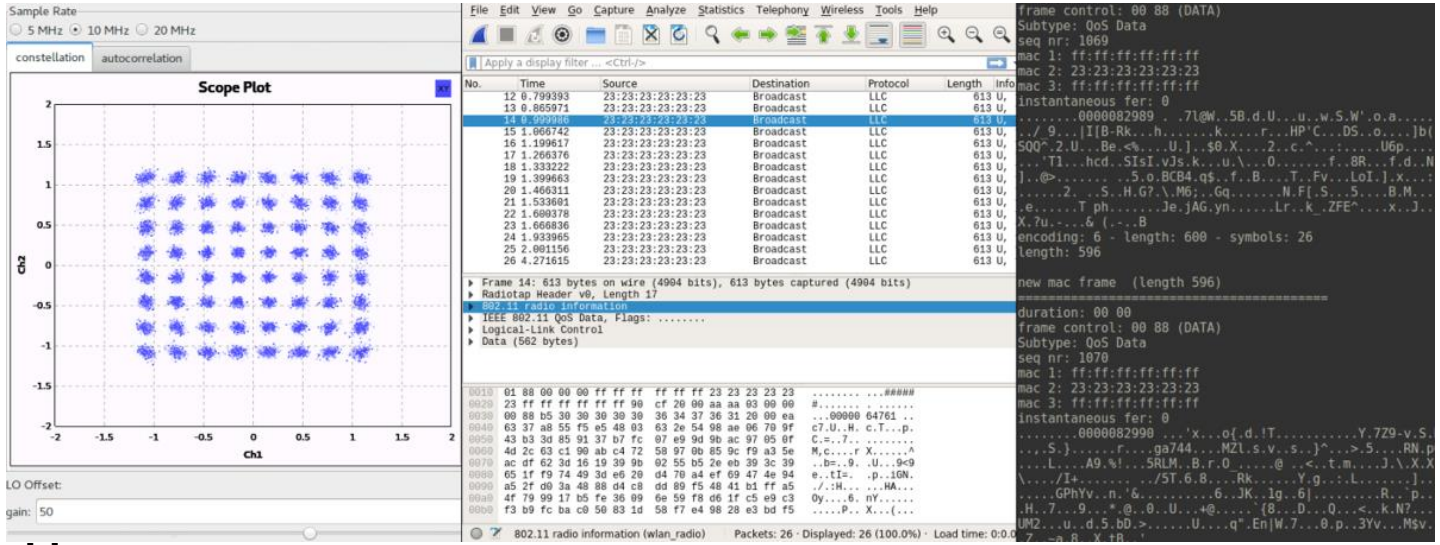
Going further

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

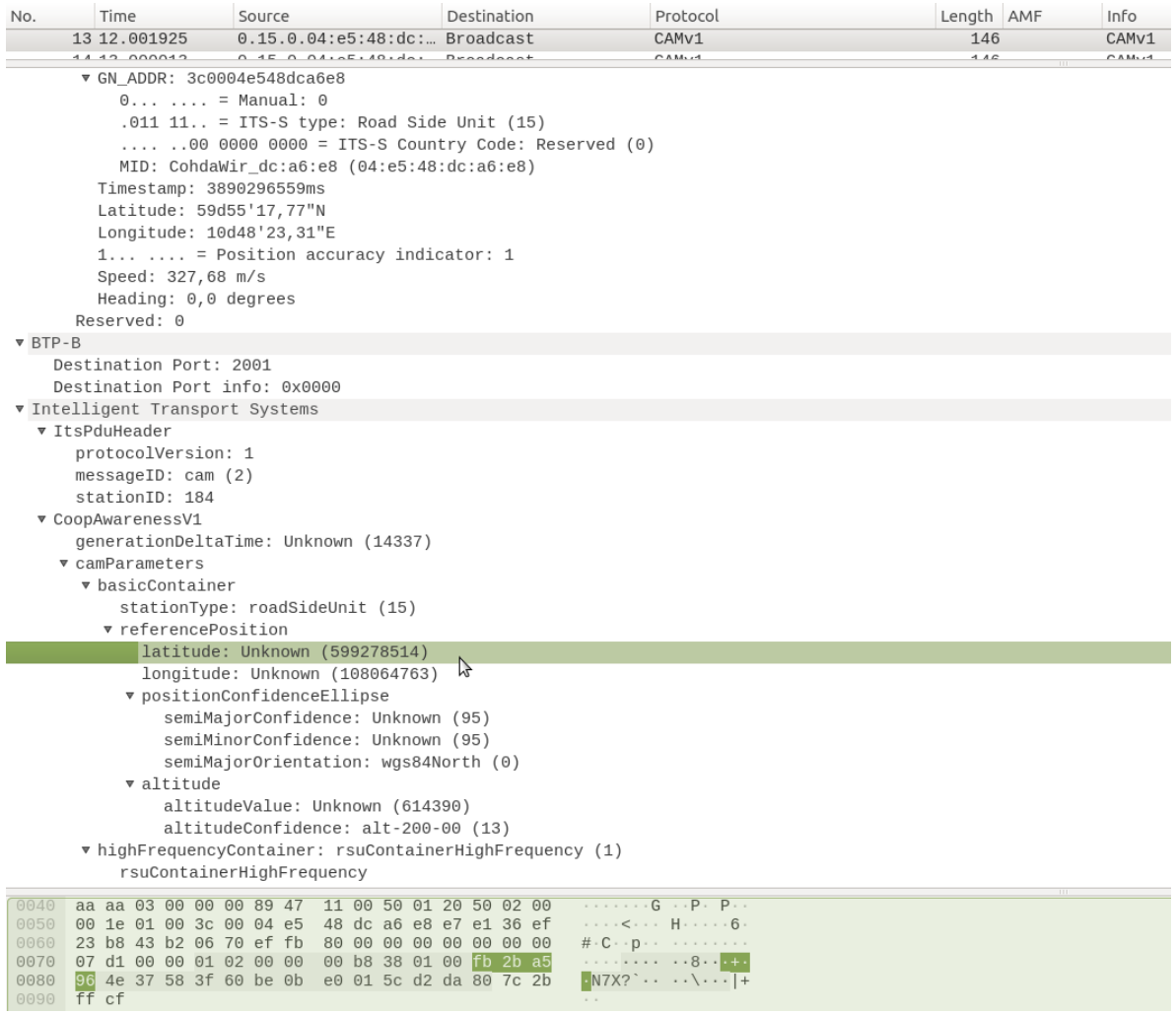
Capturing 802.11p data

- 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!):



[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

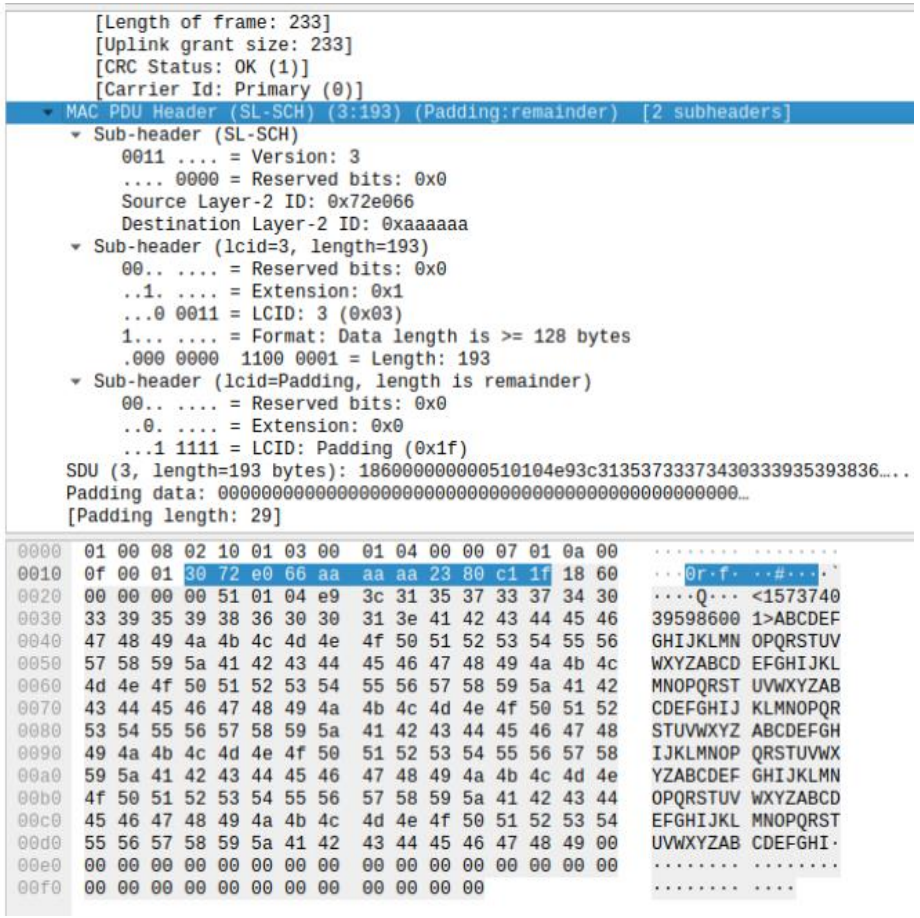


C-V2X

- 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

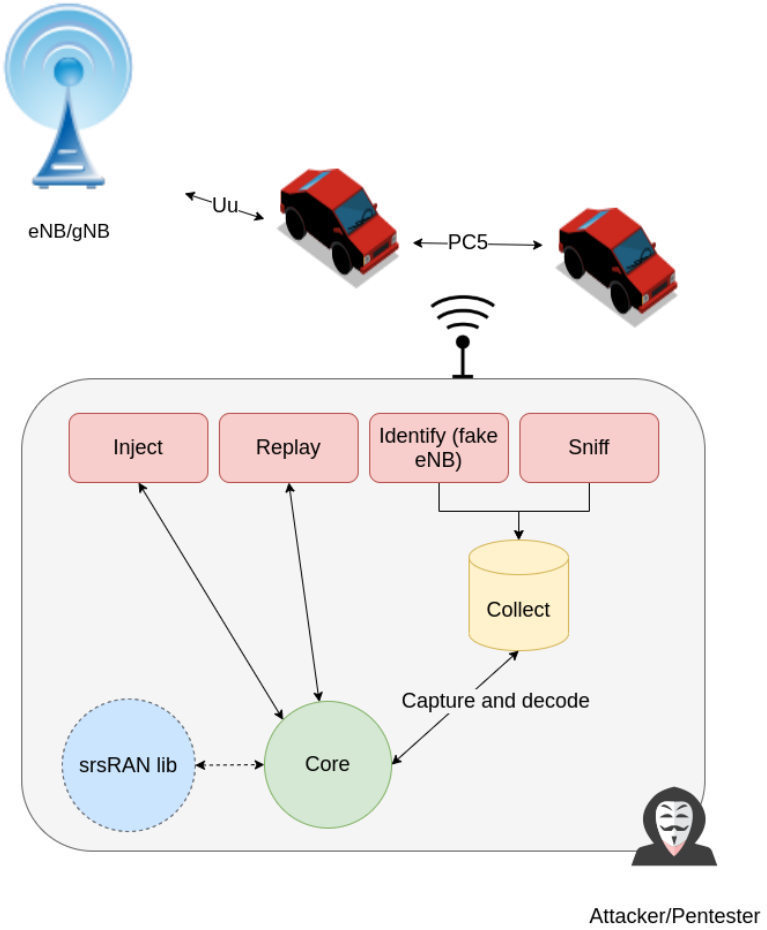
C-V2X tools and limitations

- 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



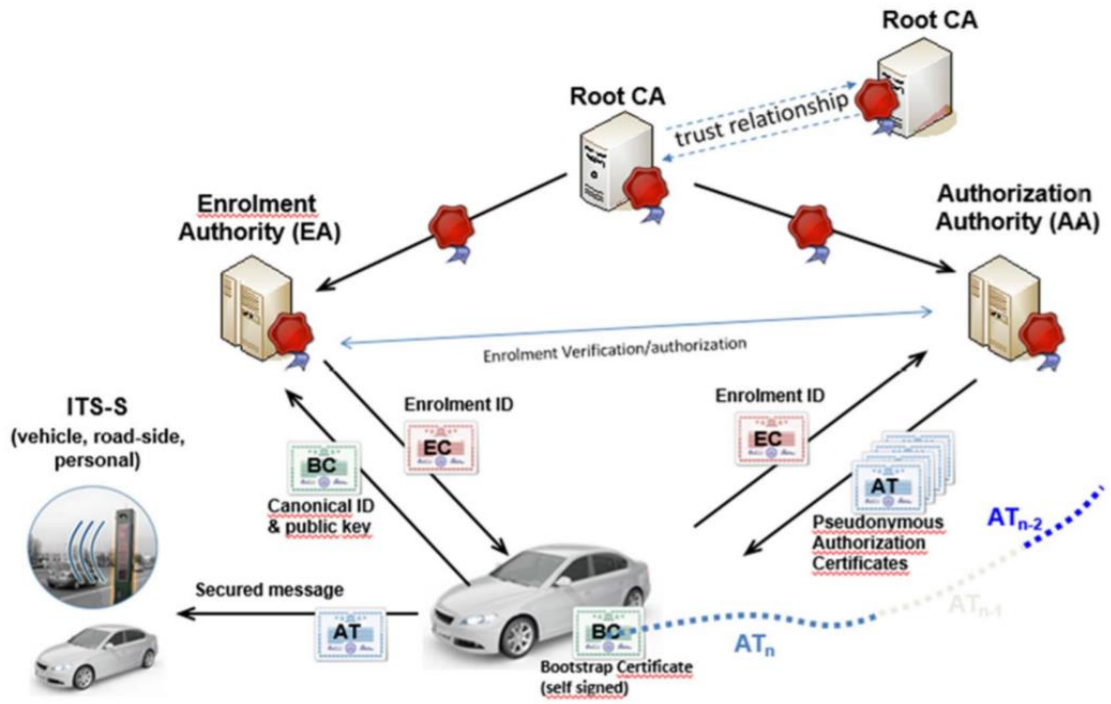
Our tool on LTE-V2X

- 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.



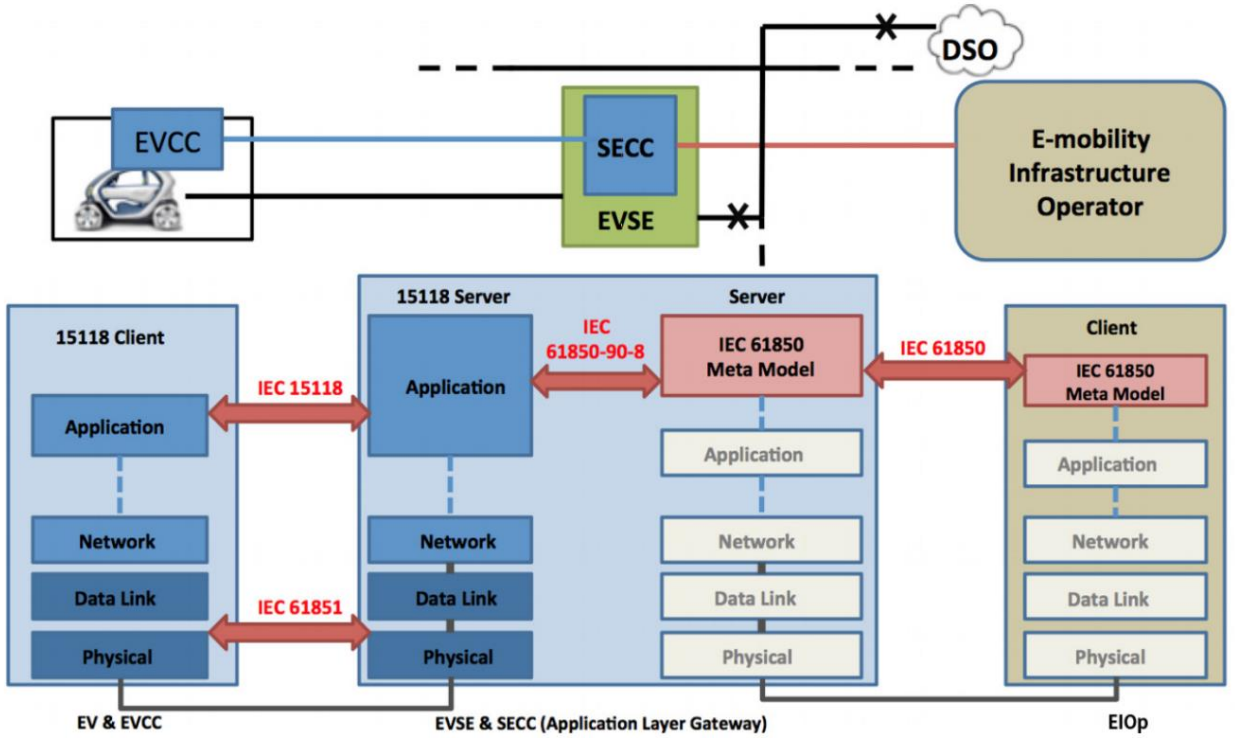
Current state of this research: Payed for some RSUs + OBUs kits during chip shortage → got scammed or getting those kits are still complicated after some years…
So we have only a simulator working, no real products to test ☹

V2V/V2I PKI: What is the real state?



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

PKI: Remember V2G?

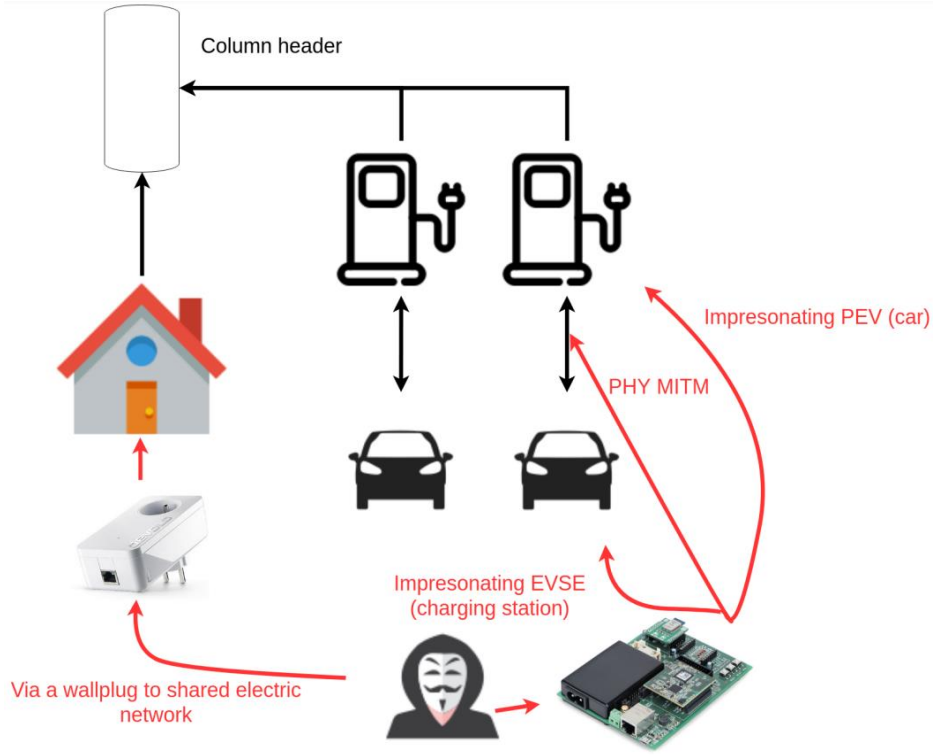


Tested solution 2019 → PKI not enforced!

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

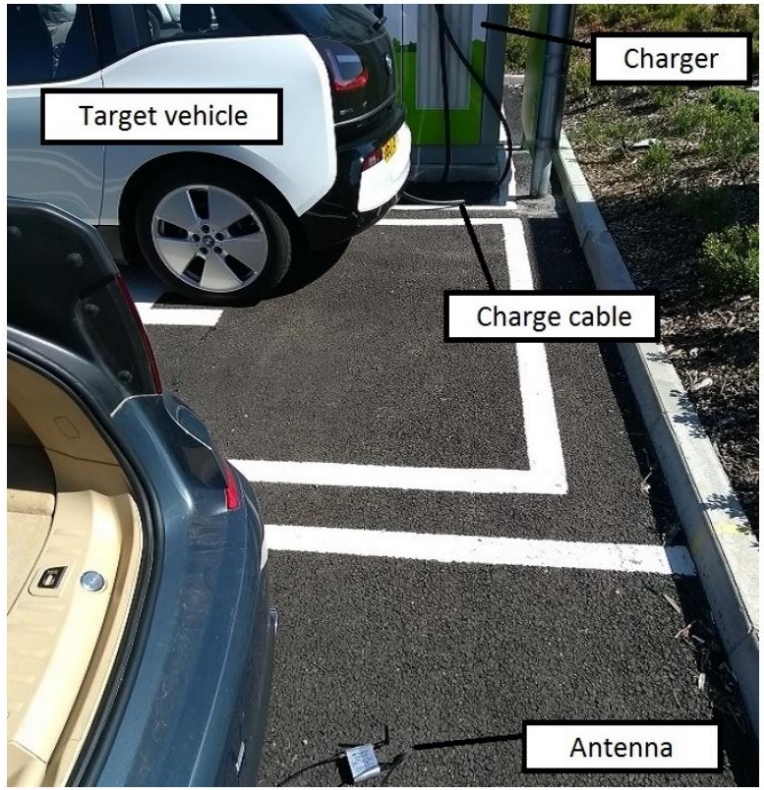
V2G flaws

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



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 communicates → 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

penthertz

Thank You

Please contact us:

contact@penthertz.com

+33 1 73 13 82 77

penthertz.com

Watch us on YouTube