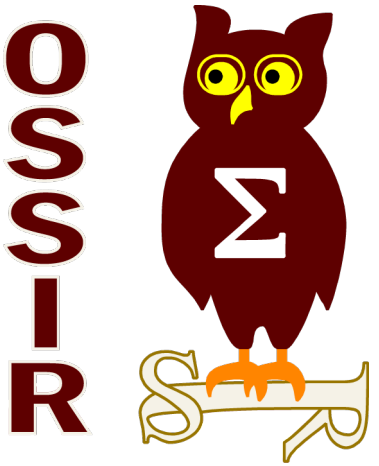
SYNACKTIV
DIGITAL SECURITY

IoT Hacking

The case of intercoms

Presented by Sébastien Dudek



About me

- **Company: Synacktiv (http://www.synacktiv.com)**
- **Twitter: @fluxius**
- **Interests: radio-communications (Wi-Fi, RFID, GSM, PLC...), networking, web, Linux security… and intercoms!**
- **Developer of http://DomzZilla.fr**
  - Engine to quickly find a housing

I’m not

- **CEH**
- **CISSP**
- **CSSP**
- **OSCP**
- **CISA**
- **CISM**
- **GIAC**
- **or whatever….**

Could be improvised to get people attention

International Information Systems Security Certification Consortium

The (ISC)² Board of Directors hereby awards

Sébastien Dudek

the credential of

Certified Information Systems Security Professional

Having met all of the certification requirements, which include the professional experience prerequisite, adoption of the (ISC)² Code of Ethics, and successful performance on the required competency examination, subject to recertification every three years, this individual is entitled to all of the rights and privileges associated with this designation, as defined in the (ISC)² Bylaws.

Chairperson

Recording Secretary

CERTIFIED INFORMATION SYSTEMS SECURITY PROFESSIONAL

CISSP®

ANSI

ISO/IEC 17024

385947

Certificate Number

December 2020

Expiration Date

Certified Since 2010

(ISC)²

and so on...

EC-Council

Certified Ethical Hacker v8

Sébastien Dudek

Congratulations! You have successfully passed the above exam. To learn more about the certification program and how this exam meets the requirements of security certification track, please visit the EC-Council web site: http://www.eccouncil.org

This test was delivered at Thomson Prometric Prime Testing Center.

8/14/2013

Date

C|EH Certified Ethical Hacker

Red team tests

- **At Synacktiv we do red team tests:**
  - spear phishing
  - web and network intrusions
  - backdoored devices
  - physical intrusions

Physical intrusions (1)

- **Enter a building to:**
  - to plug a malicious device,
  - dump computers memory,
  - or let malicious USB keys indoor, …
- **Final goal:**
  - have a foot in the targeted network
  - browse and report sensitive documents of the clients

Physical intrusion (2)

- **Main problem: we always need a way to enter to a building**
- **How?:**
  - lock-picking

    pros: subtle

    cons: sometimes hard and time consuming, break locks sometimes
  - RF attacks

    pros: subtle and clean

    cons: hard sometimes with authentication tags like MIFARE DESfire (when it's used to authenticate and not to identify...)
  - social engineering → it's all about improvisation

    pros: being natural works like a charm

    cons: leaving a trace of our face

Social engineering: a true story

- **In a public building (ERP rules):**
  - came with a “crafted” convocation
  - with the convocation → trick the reception
  - plugged the intrusion network equipment
  - quit the building
  - and an alert was raised after 3 hours

The alert

- **The client sent a picture to the boss asking: “is it one of your employees?”**

Avoid alerts or traces

- **We have to be subtle**
- **To do so:**
  - RF attacks
  - lock-picking tricks
- **Social engineering→ okay without interaction**
- **But what about attacking the intercoms?**

Intercoms today

- **Features:**
  - Pass code
  - RF tag access
  - Call a resident:

    The resident can then open the door

When calling a resident, this intercoms uses the mobile network
→ that explains the (+33)6* prefix displayed on the resident's phone

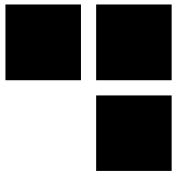
Assumptions

- **Would it be possible to play with the intercom?**
- **We tried to directly call the intercom**

  but the intercom doesn't answer to the call
- **Dump and modify the flash**

  good option, but difficult to do without being spotted in the street…
- **A mobile attack → Better!**

  but we need to understand the functioning of these intercoms first!



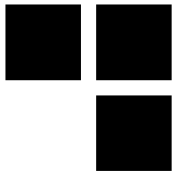
Introduction

Context

- **Intercom / door phone / house intercom**
- **A voice communication device → within a building**
- **Numeric for our case → use the mobile network (SIM/USIM cards)**
- **Allows to call a resident to identify the visitor and open a door**

Different types of intercoms exist

Different types of intercoms

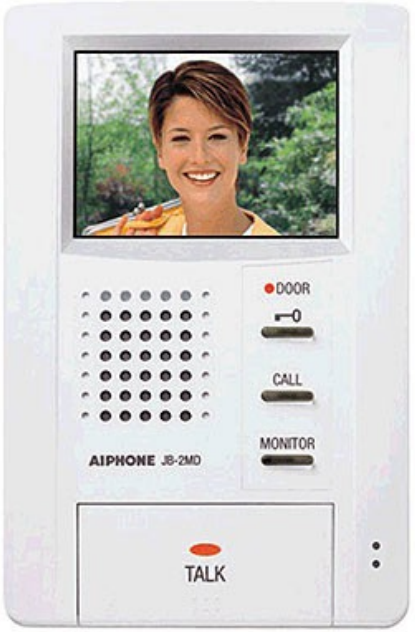


	Conventional	Simplified	Numeric
Description	Used for medium sized buildings	/	Medium sized building, or private residents
# of wires	4+n (2 for power, 2 for the door system and n → number of residents)	1 wire for power and door system + n → number of residents	Generally: no wires for each resident

Numeric intercoms

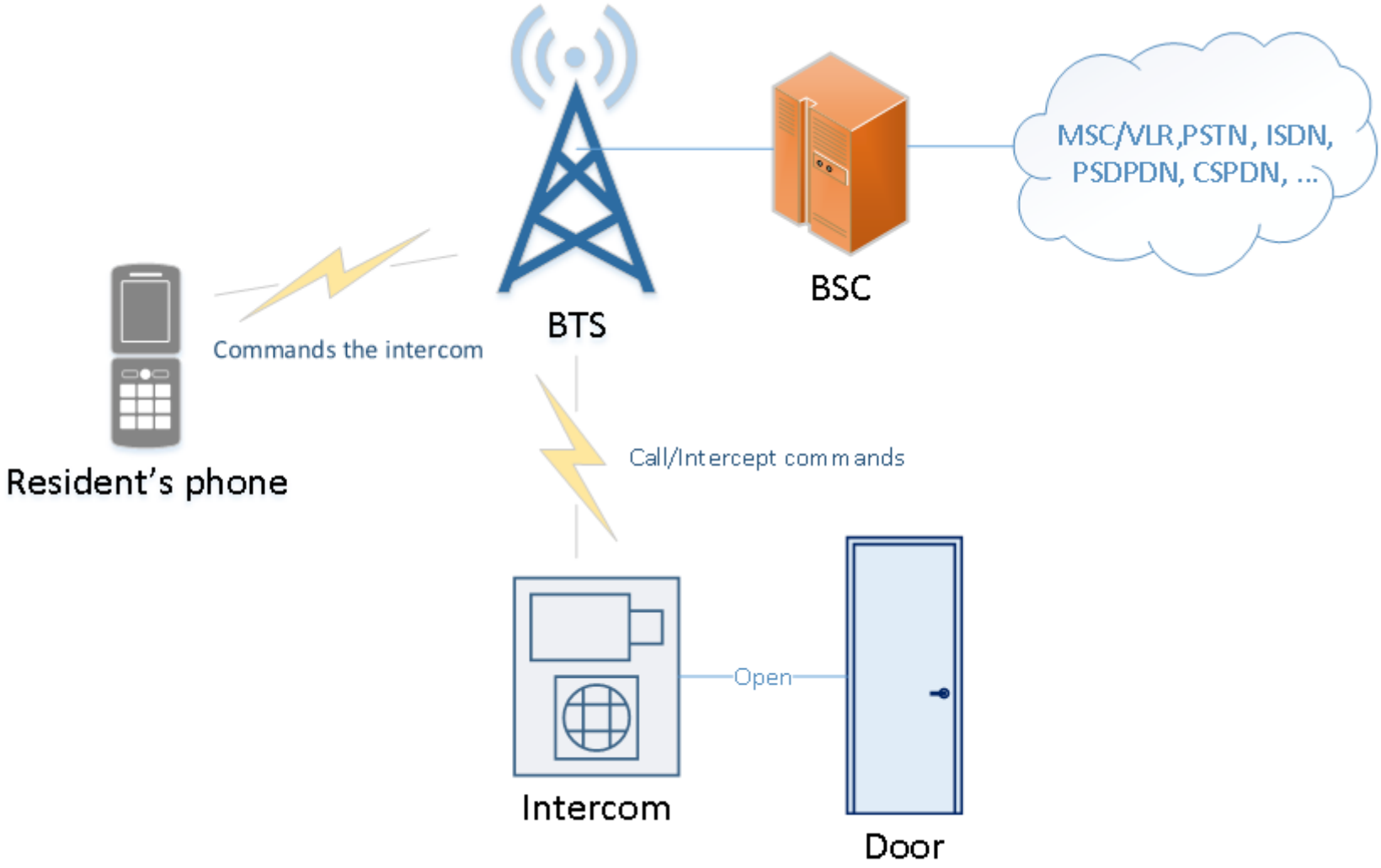
Wires replaced by:

- GSM, 3G, rarely in 4G
- or Wi-Fi…

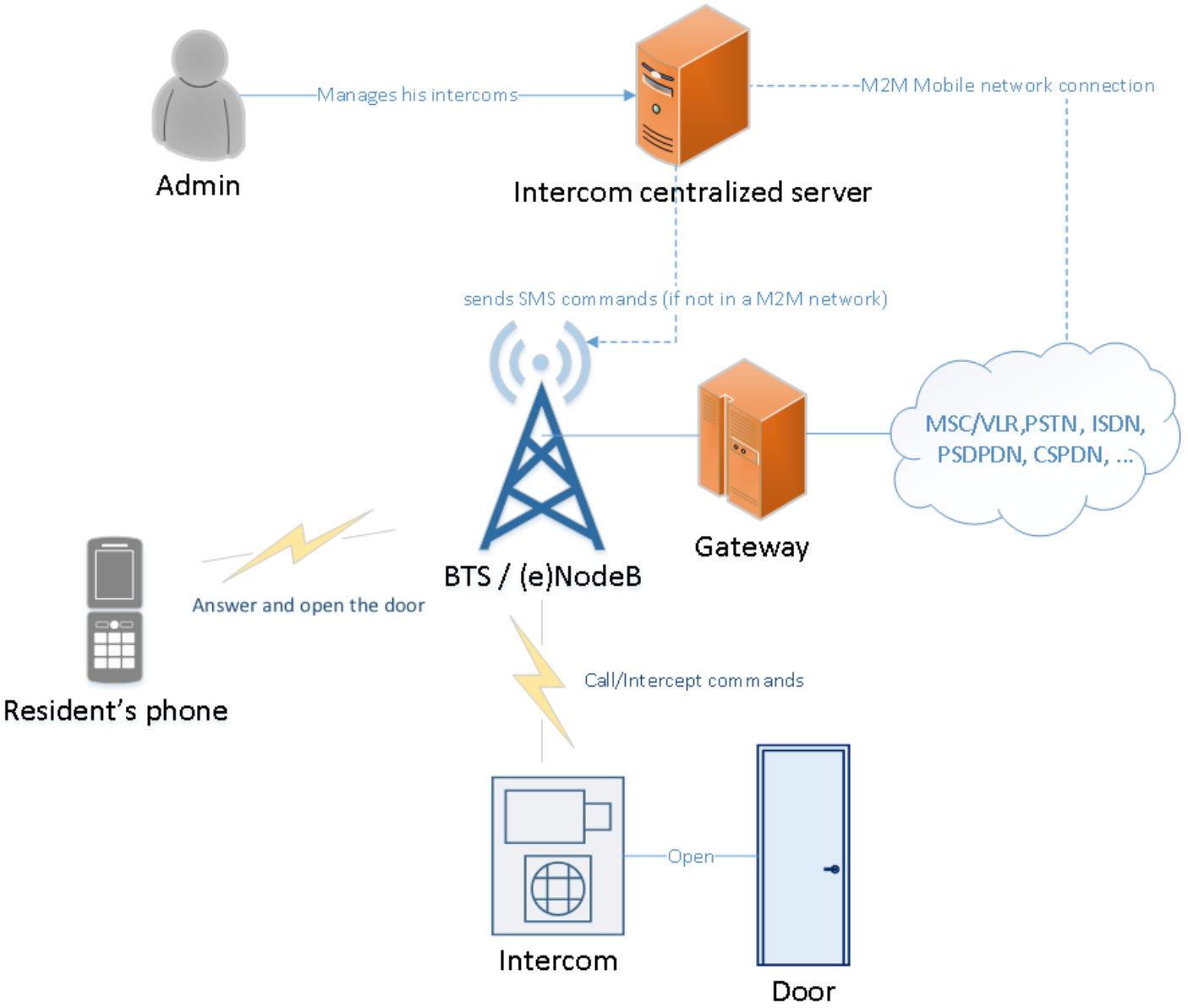


⇒ Avoid complicated and cumbersome cables
⇒ Easy installation

Numeric intercoms: simplified architecture



Network architecture with M2M



Different brands market

- **4 brands are well-known in France:**
  - Comelit
  - Intratone
  - Norasly
  - Urmet Captiv… that cost ~2000€
- **Cheaper alternatives:**
  - Linkcom → commonly used by private residents

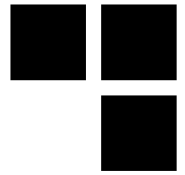
    → Our choice for our 1st analysis

How to recognize a mobile intercom

- **Not easy… maybe spotting a nice LCD screen, new stainless steel case…**
- **Or...**

Looks like a mobile module?

State Of the Art: intercoms



- **Publications about intercoms are nearly nonexistent**
- **But research on mobile security can be applied to attack these devices...**

State Of the Art: Mobile security

- **Many publications exist:**
  - **Attacks on GSM A5/1 algorithm with rainbow tables**

    (at 26c3, Chris Paget and Karsten Nohl)
  - **OsmocomBB**

    (at 2010 at 27c3, Harald Welte and Steve Markgraf)
  - **Hacking the Vodaphone femtocell**

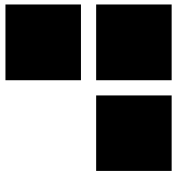
    (at BlackHat 2011, Ravishankar Borgaonkar, Nico Golde, and Kevin Redon)
  - **An analysis of basebands security**

    (at SSTIC 2014, Benoit Michau)
  - **Attacks on privacy and availability of 4G**

    (In October 2015, Altaf Shaik, Ravishankar Borgaonkar, N. Asokan, Valtteri Niemi and Jean-Pierre Seifert)
  - **How to not break LTE crypto**

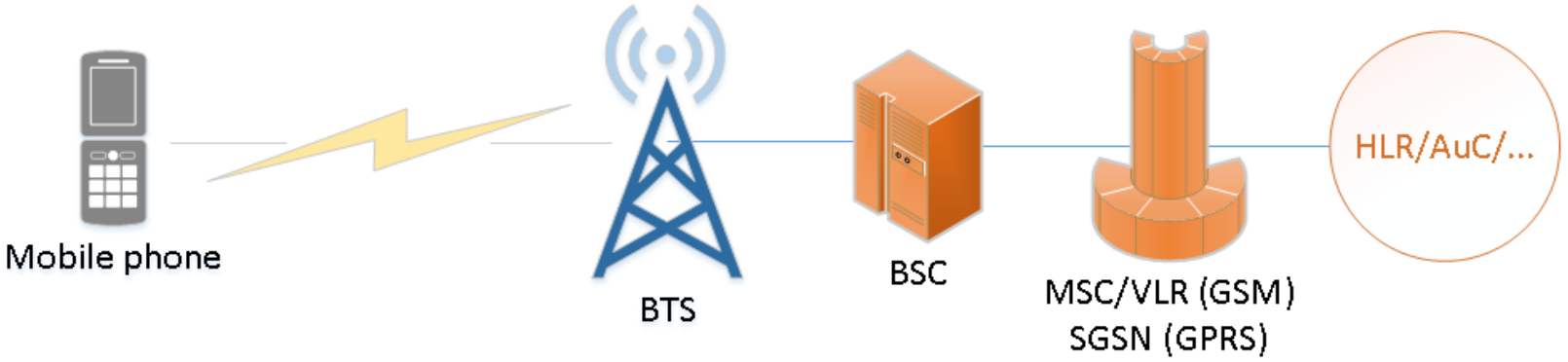
    (at SSTIC 2016, Christophe Devine and Benoit Michau)
  - And many others...

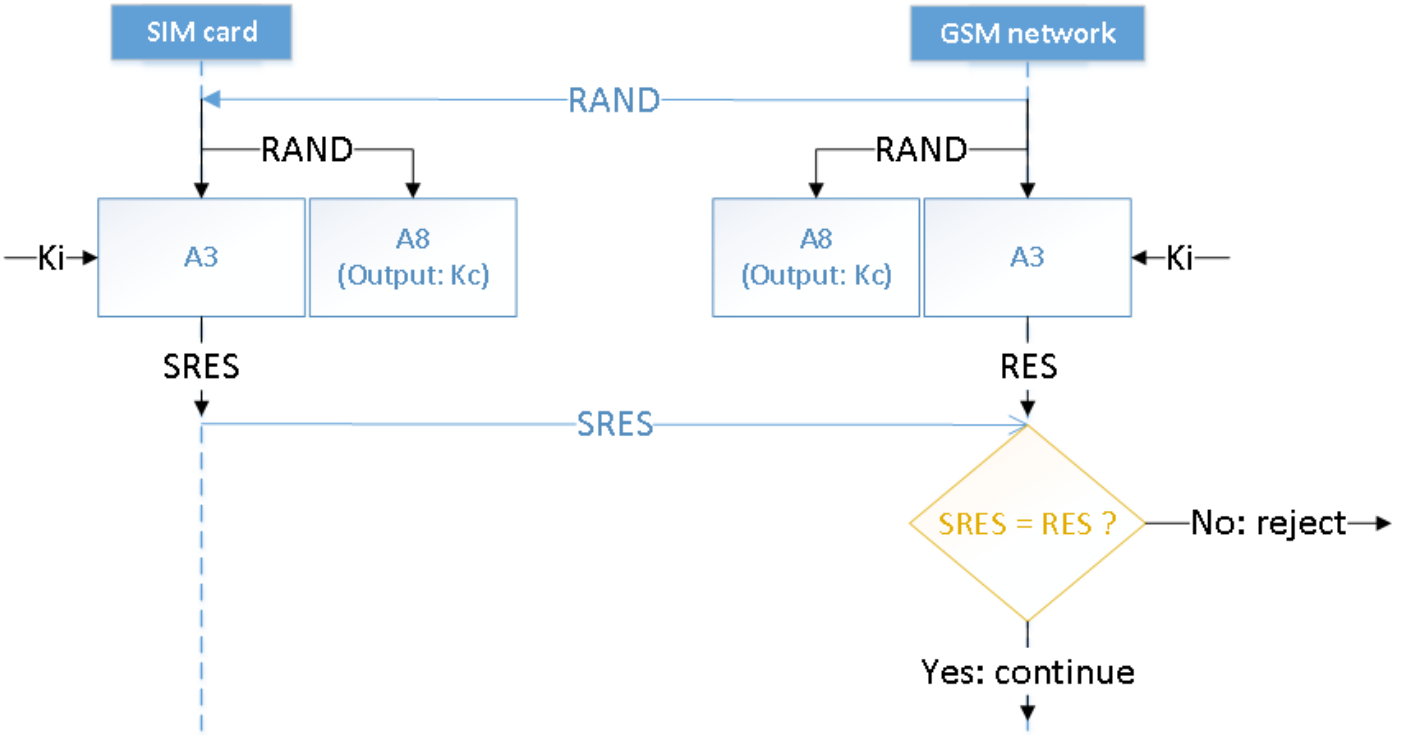
State Of the Art: tools



- **Hardware**
  - USRP from 700 € (without daughter-boards and antennas)
  - SysmoBTS from 2,000 €
  - BladeRF from 370 € (without antennas)
- **Software**
  - Setup a mobile network
    - OpenBTS: GSM and GPRS network compatible with USRP and BladeRF
    - OpenUMTS: UMTS network compatible with some USRP
    - OpenLTE: LTE network compatible with BladeRF and USRP
    - OpenAir: LTE network compatible with some USRP
    - YateBTS: GSM and GPRS network compatible with USRP and BladeRF
  - Analyze traffic
    - libmich: Analyze and craft mobile packets captured with GSMTAP
    - Wireshark: Analyze GSMTAP captured packets
    - OsmocomBB: sniff and capture GSM packets

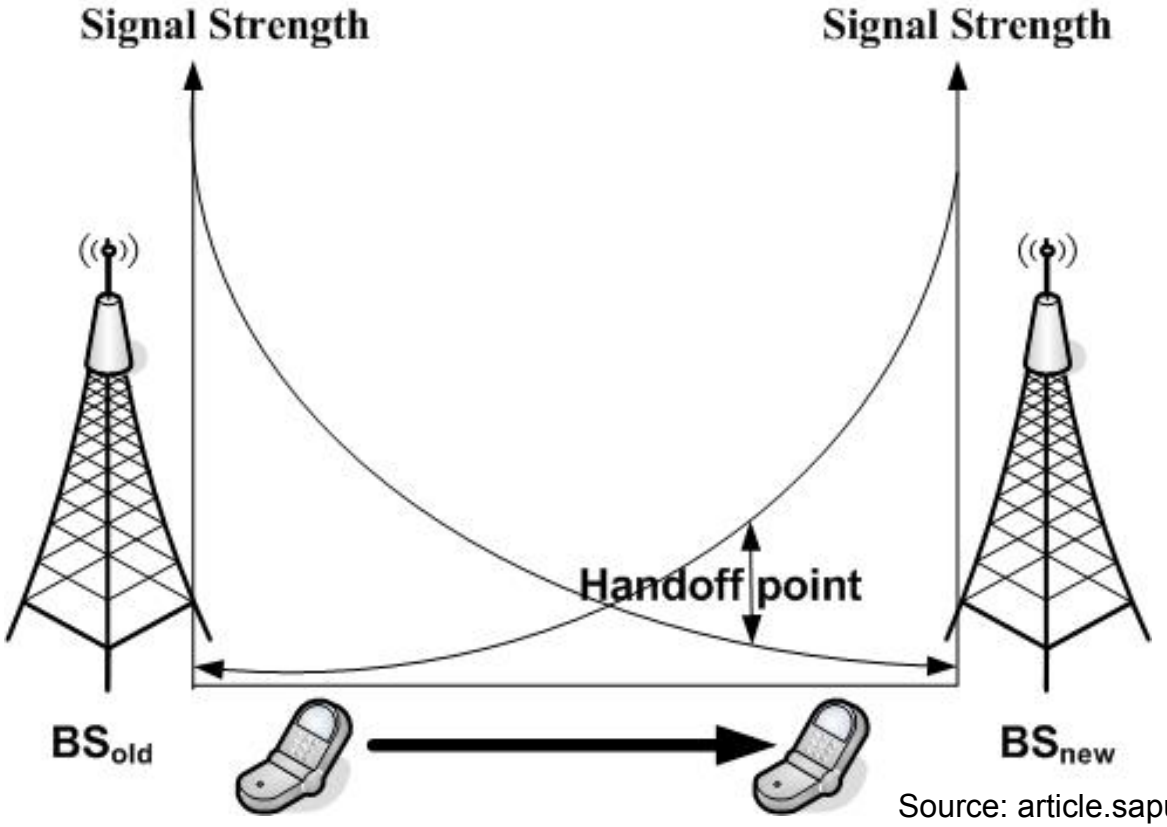
GSM and GPRS: authentication





- BTS: Base Transceiver Station
- BSC: Base Station Controller
- MSC: Mobile Switch Center
- VLR: Visitor Location Register
- HLR: Home Location Register
- AuC: Authentication Center

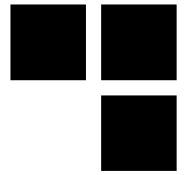
GSM and GPRS: Handover



Source: article.sapub.org

A stronger signal will likely attract User Equipments
→ Useful for attackers

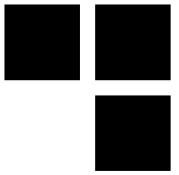
GSM and GPRS: possible attacks



- **No mutual authentication → Fake rogue BTS**
- **Reuse of Authentication triplet RAND, RES, K_c many times**
- **Signaling channel not encrypted → open for attacks**
- **Attacks on the A5/1 algorithm**

⇒ Interception is possible on GSM and GPRS

3G/4G: advantages



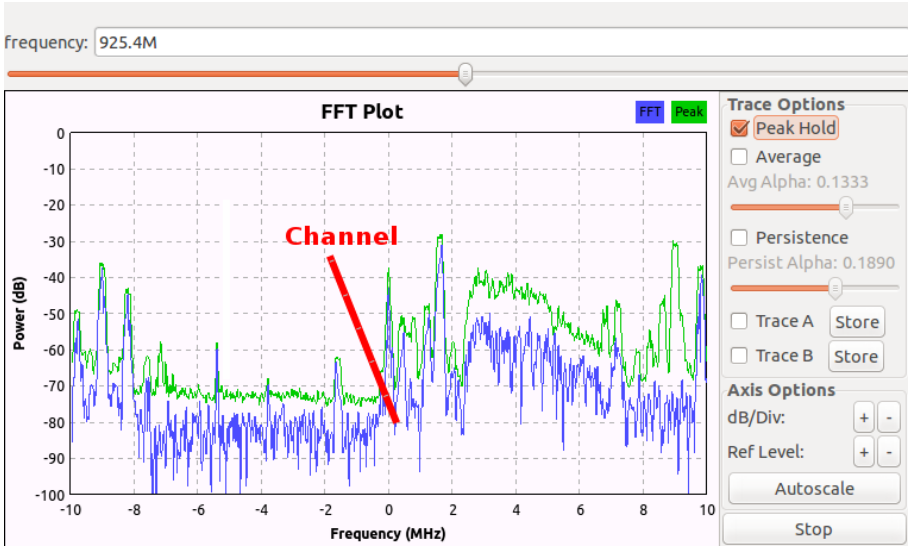
	GSM	3G	4G
Client authentication	YES	YES	YES
Network authentication	NO	Only if USIM is used (not SIM)	YES
Signaling integrity	NO	YES	YES
Encryption	A5/1	KASUMI | SNOW-3G	SNOW-3G | AES | ZUC...

Mobile interception: signal attraction

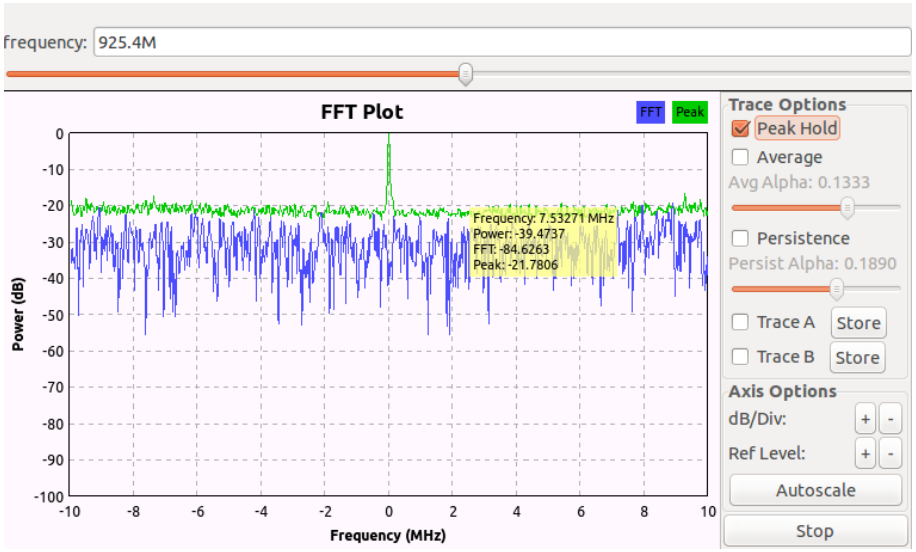
- **A User Equipment connects to the closest Base Station**

- **3G/4G downgrades to 2G via**
  - protocol attacks → difficult
  - jamming attacks → a simple Gaussian noise in targeted channels

Jamming is generally basic...

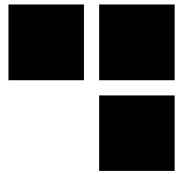


Before



After

The 3G module



- **Found in a public documentation:**

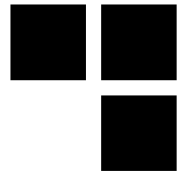
« Lorsque le réseau 3G est inexistant sur les lieux de l'installation, le bloc 3G cherchera le réseau GSM automatiquement et pourra résumer ses fonctionnalités dans ce mode :
- Appel Audio (sans Visio).
- Mise à jour en temps réel sur le réseau GSM et non plus 3G. »

= If 3G is unreachable → use 2G instead!

To jam a 3G channel

- **We can buy a jammer + disable 2G Tx**
- **Or for each operator:**
  - enumerate the list of close UARFCN (UTRA Absolute Radio Frequency Channel Number)
  - with UARFCN → translate into central frequencies to jam the channels
  - send Gaussian noise into each detected channel using SDR

How to enumerate UARFCN? (1)

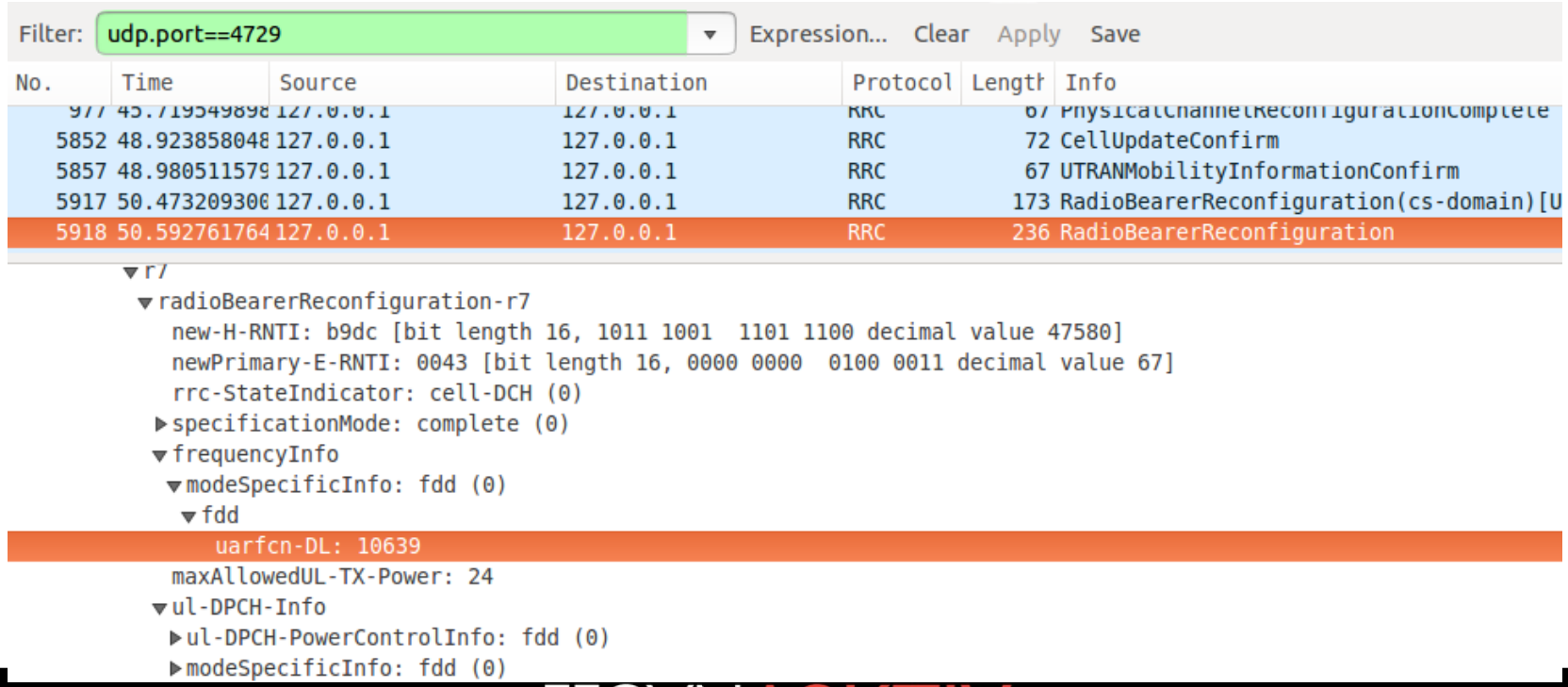


- **OsmocomBB only works for GSM =(**

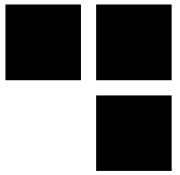
```
OsmocomBB# show cell 1
ARFCN  |MCC    |MNC    |LAC    |cell ID|forb.LA|prio   |min-db |max-pwr|rx-lev
-------+-------+-------+-------+-------+-------+-------+-------+-------+-------
   1   |208    |01     |0x     |0xe    |n/a    |n/a    |-110   |   5   |-71
   3   |208    |01     |0x     |0xb    |n/a    |n/a    |-110   |   5   |-76
   7   |208    |01     |0x     |0xa    |n/a    |n/a    |-110   |   5   |-74
  11   |208    |01     |0x     |0xe    |n/a    |n/a    |-110   |   5   |-75
  77   |208    |10     |0x     |0x9    |no     |normal |-105   |   5   |-84
 513DCS|208    |01     |0x     |0xd    |n/a    |n/a    | -95   |   0   |-82
 518DCS|208    |01     |0x     |0x5    |n/a    |n/a    | -95   |   0   |-79
 609DCS|208    |01     |0x     |0xf    |n/a    |n/a    | -95   |   0   |-70
 744DCS|208    |10     |0x     |0xe    |n/a    |n/a    | -95   |   0   |-91
 976   |208    |20     |0x     |0xc    |n/a    |n/a    |-104   |   5   |-81
 978   |208    |20     |0x     |0xc    |n/a    |n/a    |-104   |   5   |-79
 979   |208    |20     |0x     |0x0    |n/a    |n/a    |-104   |   5   |-84
 982   |208    |20     |0x     |0xc    |n/a    |n/a    |-104   |   5   |-74
 984   |208    |20     |0x     |0xc    |n/a    |n/a    |-104   |   5   |-57
 986   |n/a    |n/a    |n/     |n/a    |n/a    |n/a    |n/a    |n/a    |n/a
1011   |208    |20     |0x     |0x9    |n/a    |n/a    |-104   |   5   |-87
1012   |208    |20     |0x     |0xb    |n/a    |n/a    |-104   |   5   |-84
```

Baseband diag interfaces (1)

- **Android phones with a XGold baseband → *`/dev/ttyACM0`* → use x*goldmon* tool**
- **UMTS RRC (Radio Resource Control) messages → get DL UARFCN**

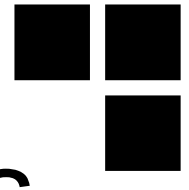


Baseband diag interfaces (2)



- **Qualcomm basebands sometimes expose a *`/dev/diag`* interface that could be exploited**
- **But a universal (and dirty) method exists with Samsung mobiles**

Cheap and dirty UARFCN enumerator with Samsung Mobiles



- **When entering the ServiceMode (e.g: **#0011#*) in Samsung and trying to register**

  → the DL and UL UARFCN are logged in logcat

- **We can parse the logcat output to get the UARFCN**

```
[...]
LOG:>>[HIGH]oemtestmode.c,403,Idle: dl_uarfcn 10688
ul_uarfcn 9738<<
[...]
```

With our *mobiletools* framework

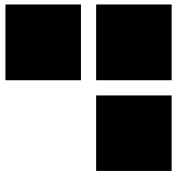
- **Our development → tool that monitor 3G and 4G cells with a cheap Samsung device**

```
~# ./ServiceMode.py
=> Requesting a list of MCC/MNC
[+] List of operators:
{'F-Bouygues Telecom': ['20820'], 'Orange F': ['20801'], 'F SFR': ['20810'], 'Free':
['20815']}
=> Changing MCC/MNC for: 20810
[+] New cell detected [Cell-ID/TAC (30***)]
 Network type=4G
 PLMN=208-20
 Band=3
 Downlink EARFCN=1850
[+] New cell detected [Cell-ID/TAC (46***)]
 Network type=4G
 PLMN=208-10
 Band=3
 Downlink EARFCN=1850 [...]
```

Downgrade 3G → 2G demo

- **Targeted channels jamming**
- **Using a simple HackRF for ~300€**

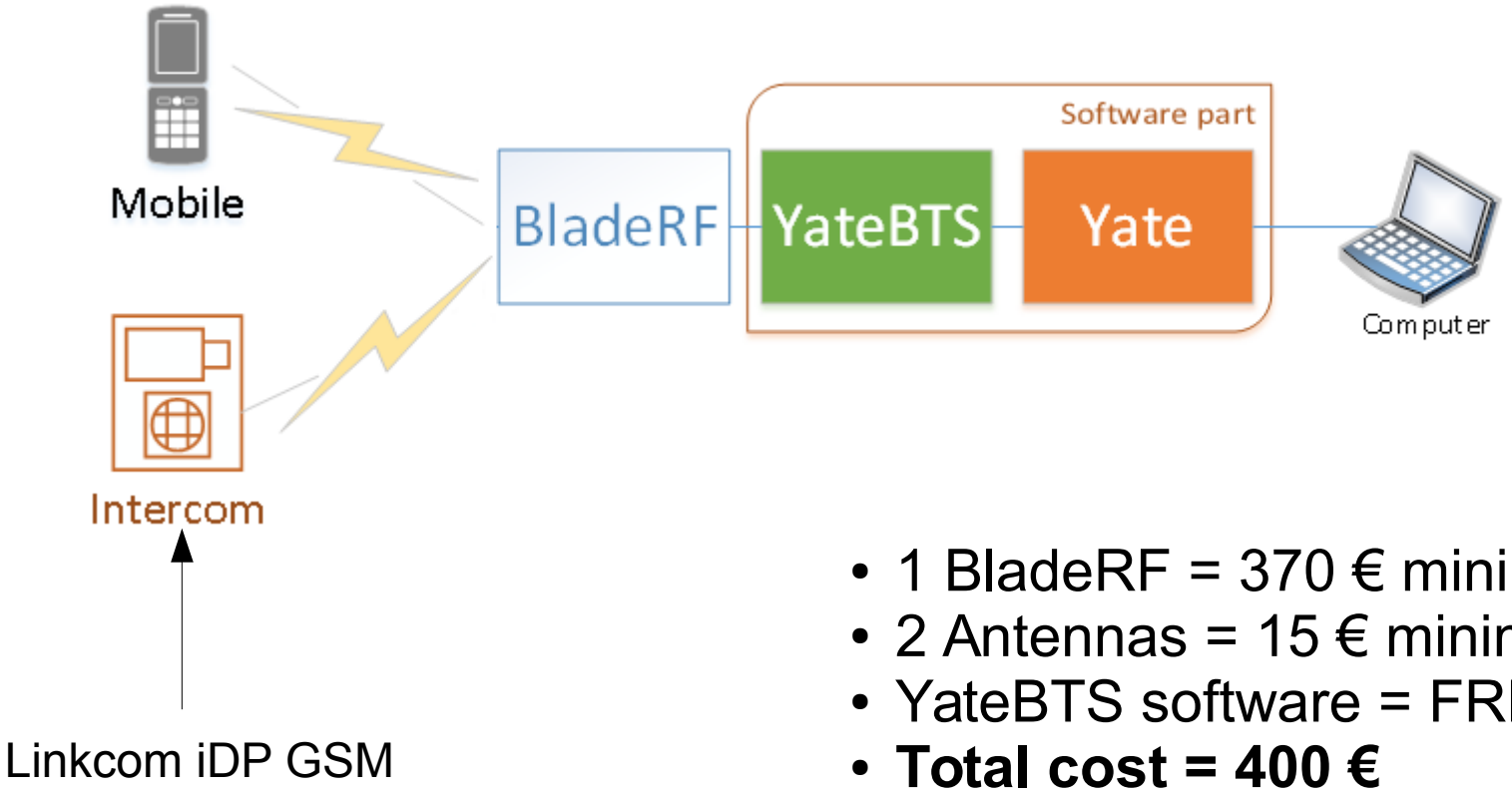
  → works also with a USRP (~700€), or a bladeRF (~400€)



Attacking our first intercom

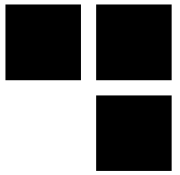
GSM Lab setup: for interception

No full duplex with hackRF → we use a bladeRF instead!



- 1 BladeRF = 370 € minimum
- 2 Antennas = 15 € minimum each
- YateBTS software = FREE
- **Total cost = 400 €**

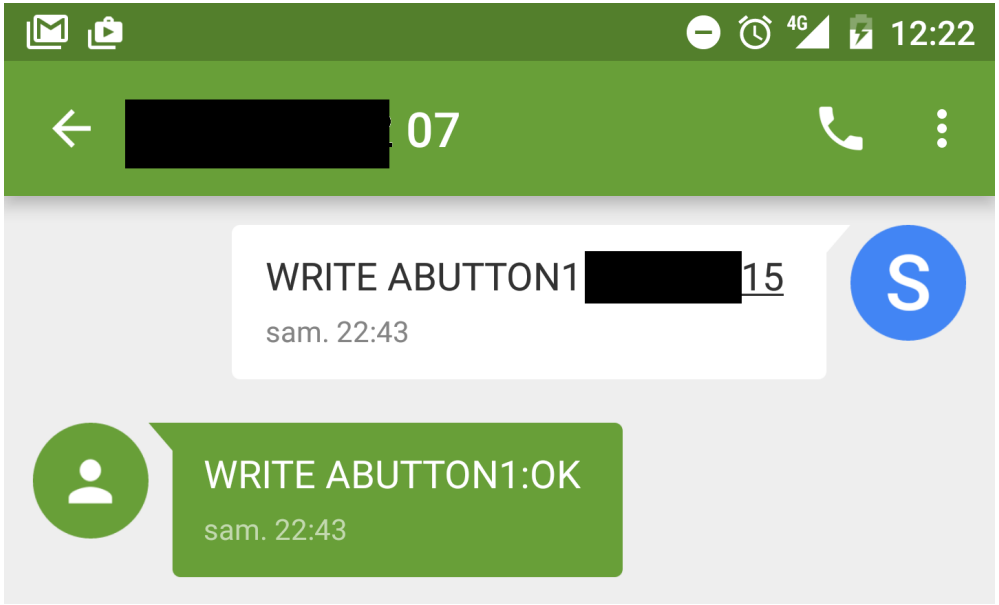
Intercom setup: configuration



- **This intercom can be configured in 3 ways:**
  - With a programming interface and the Link iDP manager software
  - With a SIM card reader/programmer
  - **Via SMS messages**
- **The SIM card is used as a memory → contains all the settings**
- **A first administrator number “ADMIN1” has to be setup in the SIM card contacts**

First impressions

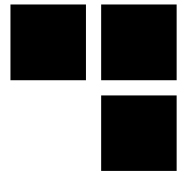
- **Our goals:**
  - impersonate a number, or find a way to bypass it
  - then open a door, or send commands to the intercoms
  - …
- **A good indicator → after sending commands, an acknowledgment is performed by SMS**



Hypotheses as a potential attacker

- **We don’t know the mobile operator**
- **We don’t know intercom’s number**
- **The commands could be found with public or leaked documentations, or by performing a firmware analysis**

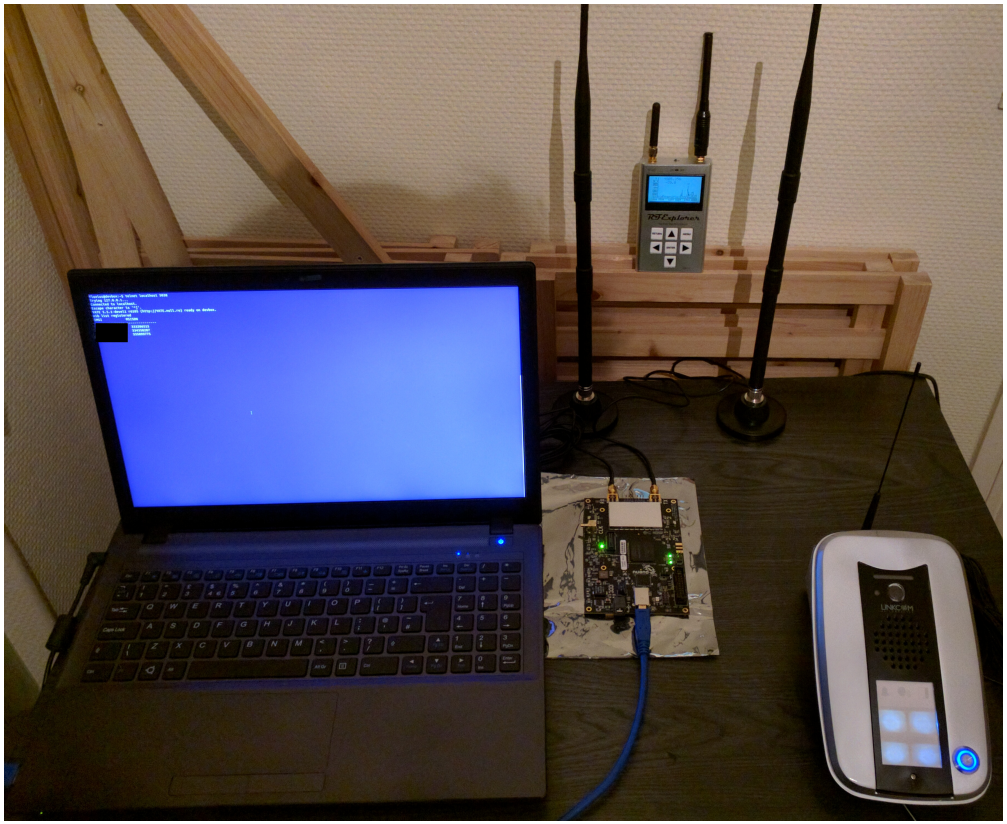
Attacker steps to open the door



1. **Recognize intercom's operator to trap it**
2. **Leak, or guess, numbers to impersonate**
3. **Register my phone with the leaked resident number on the fake BTS**
4. **Call myself**
5. **Open the door!**

To trap the intercom

- **Bruteforcing the 4 MCC/MNC (FR)**
  - 15min~ waiting for each MCC/MNC
- **Strong GSM signal**
- **Button push → calling intercepted → success!**



Note: The used MCC/MNC but mostly the used channel can be discovered with jamming tests over the different channels.

What's next? Let's leak numbers!

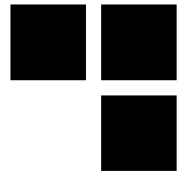
- **Activate GSM tapping on YateBTS → Wireshark**
- **Then push on buttons → CC SETUP**

```
84933 406.0349243…  127.0.0.1   127.0.0.1   LAPDm   81 I, N(R)=1, N(S)=0(DTAP) (CC) Setup
84935 406.0384471…  127.0.0.1   127.0.0.1   LAPDm   81 S, func=RR, N(R)=1
84947 406.0571079…  127.0.0.1   127.0.0.1   LAPDm   81 I, N(R)=1, N(S)=1(DTAP) (CC) Call Proceeding
84955 406.0582432…  127.0.0.1   127.0.0.1   LAPDm   81 U, func=UI
84966 406.0760920…  127.0.0.1   127.0.0.1   LAPDm   81 U, func=UI

    GSM Frame Number: 0
    Channel Type: FACCH/F (9)
    Antenna Number: 0
    Sub-Slot: 0
  Link Access Procedure, Channel Dm (LAPDm)
    Address Field: 0x01
    Control field: I, N(R)=1, N(S)=0 (0x20)
    Length Field: 0x49
  GSM A-I/F DTAP - Setup
    Protocol Discriminator: Call Control; call related SS messages (3)
      .... 0011 = Protocol discriminator: Call Control; call related SS messages (0x03)
      0... .... = TI flag: allocated by sender
      .000 .... = TIO: 0
    01.. .... = Sequence number: 1
    ..00 0101 = DTAP Call Control Message Type: Setup (0x05)
    Called Party BCD Number - (       515)
      Length: 6
      1... .... = Extension: No Extension
      .000 .... = Type of number: unknown (0x00)
      .... 0001 = Numbering plan identification: ISDN/Telephony Numbering (ITU-T Rec. E.164 / ITU-T Rec. E.163) (0x01)
      Called Party BCD Number:       515

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E.
0010  00 43 f7 4d 40 00 40 11  45 5a 7f 00 00 01 7f 00   .C.M@.@. EZ......
0020  00 01 97 fc 12 79 00 2f  fe 42 02 04 01 04 40 00   .....y./ .B....@.
0030  00 00 00 00 00 00 09 00  00 00 01 20 49 03 45 04   ........ ... I.E.
0040  06 60 04 02 00 05 81 5e                 5 f5 2b   .`.....       ..+
0050  2b                                                +
```

What’s next? Let's open the door!



- **Leaked number → affect it to your IMSI in *tmsidata.conf***

```
[tmsi]
last=007b0005
[ues]
20820<attacker’s IMSI>=007b0003,35547XXXXXXXXXX,
<resident or admin number>,1460XXXXXX,
ybts/TMSI007b0003
# associating attacker IMSI with a resident number
[...]
```

What’s next? Let's backdoor it!

- **Leak the admin number:**
  - buttons (call, or alarm triggers, etc.)
  - social engineering
- **Find commands:**
  - public or leaked documentations
  - Passive channel monitoring → good luck!
  - or buy the same model in commercial web sites such as “leboncoin”, eBay, and so on.
- **In our case with Linkcom iDP:**

Command	Description
READ <NAME>	Read the number of a button, or an admin (ADMIN[1-9]).
WRITE <NAME> <number>	Add or update a number associated to a name.
CAL AT<command suffix>	Send an AT command to the baseband through SMS!

AT commands?

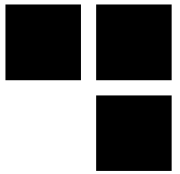
- **We can interact with Intercom's baseband:**
  - retrieve SMS messages → *AT+CMGL="ALL"*
  - spying building door conversations with auto-answer feature (if not disabled) → *ATS0=1*
  - and so on.

Demo

- **Trapping an intercom**
- **Impersonating a resident**

Call premium rate numbers

- **We can modify a contact → why not choose a premium number?**
  - Allopass
  - Optelo
  - Hipay
  - and so on.

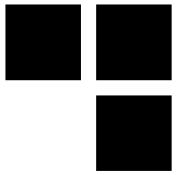


Attacking M2M intercoms

Our sample

- **Pretty the same as this one:**

Intercoms using M2M SIM/USIM cards



- **Provided with a M2M SIM/USIM card**

  → more than 10 years subscription

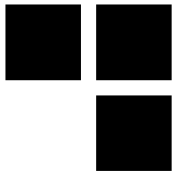
  → the mobile operator provides a virtual network to manage the intercoms

- **Use the UMTS network by default**

  → GSM is used if UMTS is unreachable

- **Intercoms → managed by a centralized server**

  → It's an interesting new vector of attacker, but there are many others...

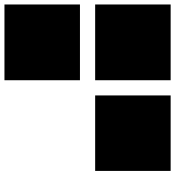


Quick look on radio frequency features

RF features

- **To enter into the building, a resident can also:**
  - use the pass code
  - but also a RF tag and or push a button from a same remote case

RF tag



- **The tag is shown as a ISO/IEC 14443 Type A tag → like MIFARE classic it's a Vigik token**

```
# nfc-list
[...]
1 ISO14443A passive target(s) found:
ISO/IEC 14443A (106 kbps) target:
    ATQA (SENS_RES): 00  04
       UID (NFCID1): eb  34  **  **
      SAK (SEL_RES): 08
```

- **Vigik token of a resident can be cloned**
- **But people are still interested to have a multipass tag → be able to break 1024-bit RSA keys → see the state of the art by Renaud Lifchitz at Hackito Ergo Sum 2014**

Remote control

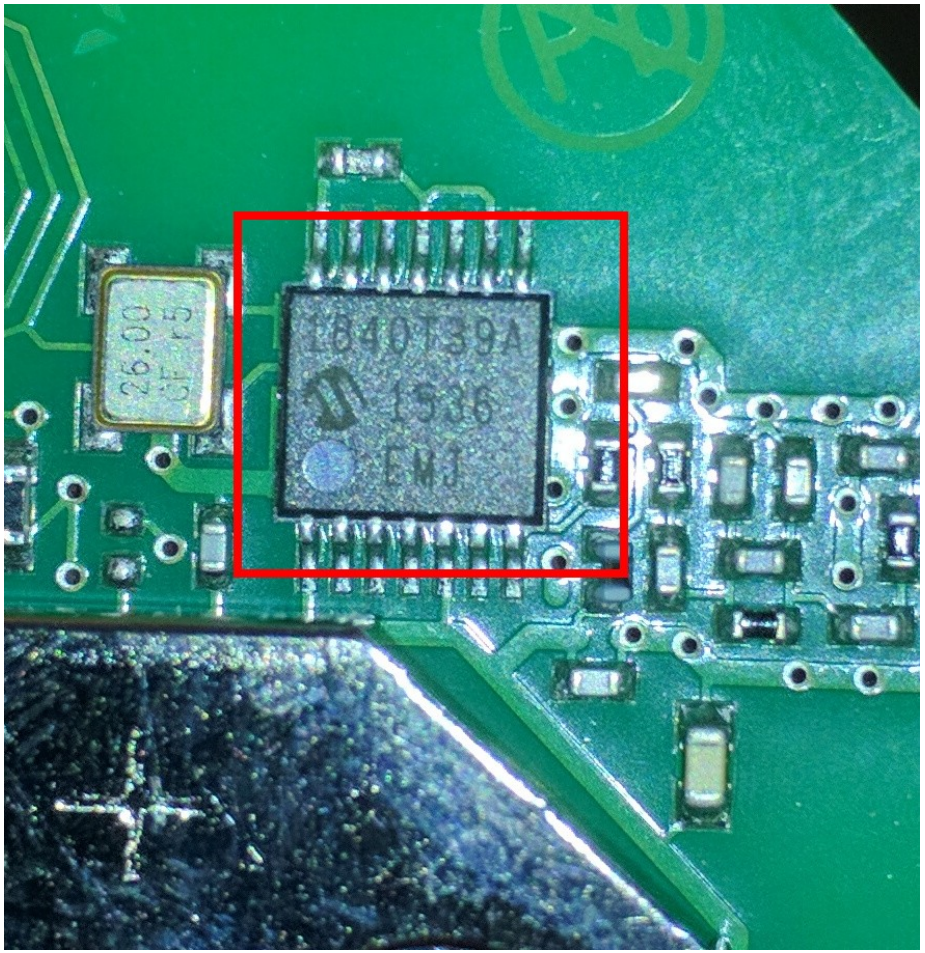
- **The remote control use RF frequencies to send a signal → open the door**
- **To capture the signal we have two approach:**
  - enumerating frequencies and catch the strange signal when hitting the remote button
  - or open the remote's case

Deep within the remote

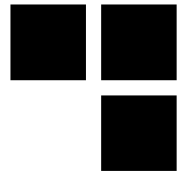
- **Includes:**
  - a Vigik paper tag
  - and a RF transmitter

Zoom on the transmitter

- **We can read its identifier: 1840T39A**



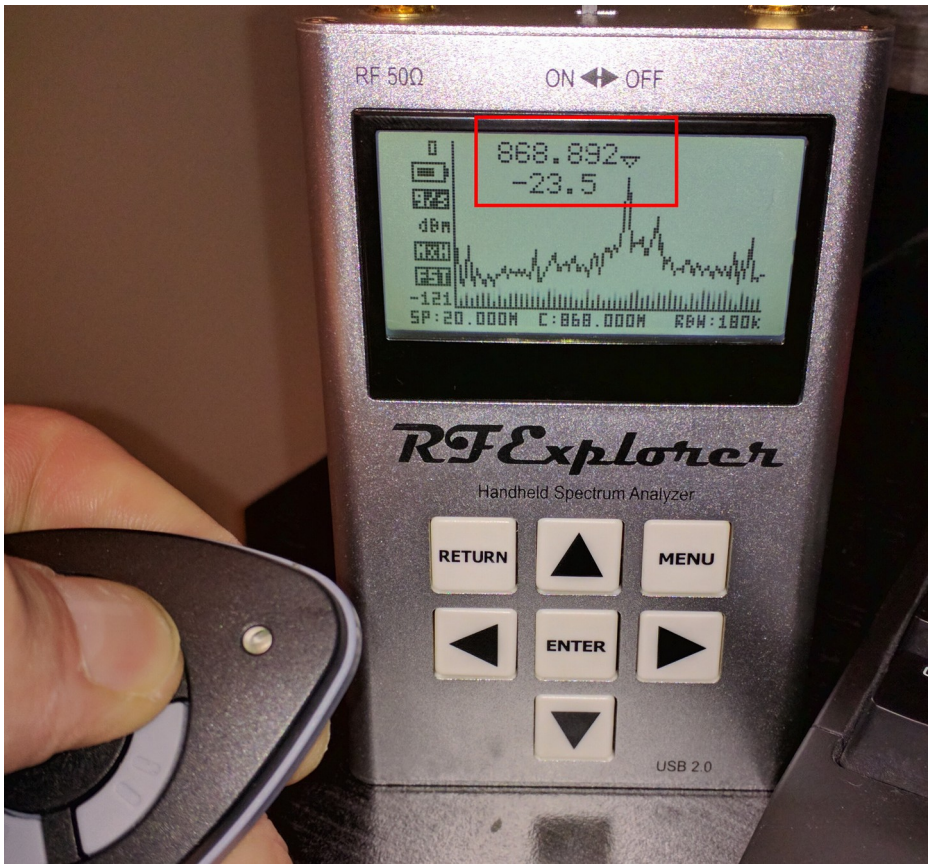
1840T39A specifications



- **Public:**
  http://ww1.microchip.com/downloads/en/DeviceDoc/40001636B.pdf
- **Supported frequency bands: 310, 433, 868 and 915 MHz**
- **Supported modulations: FSK (up to 100 kbps) and OOK (up to 10 kbps)**
- **Let's identify the working frequency**

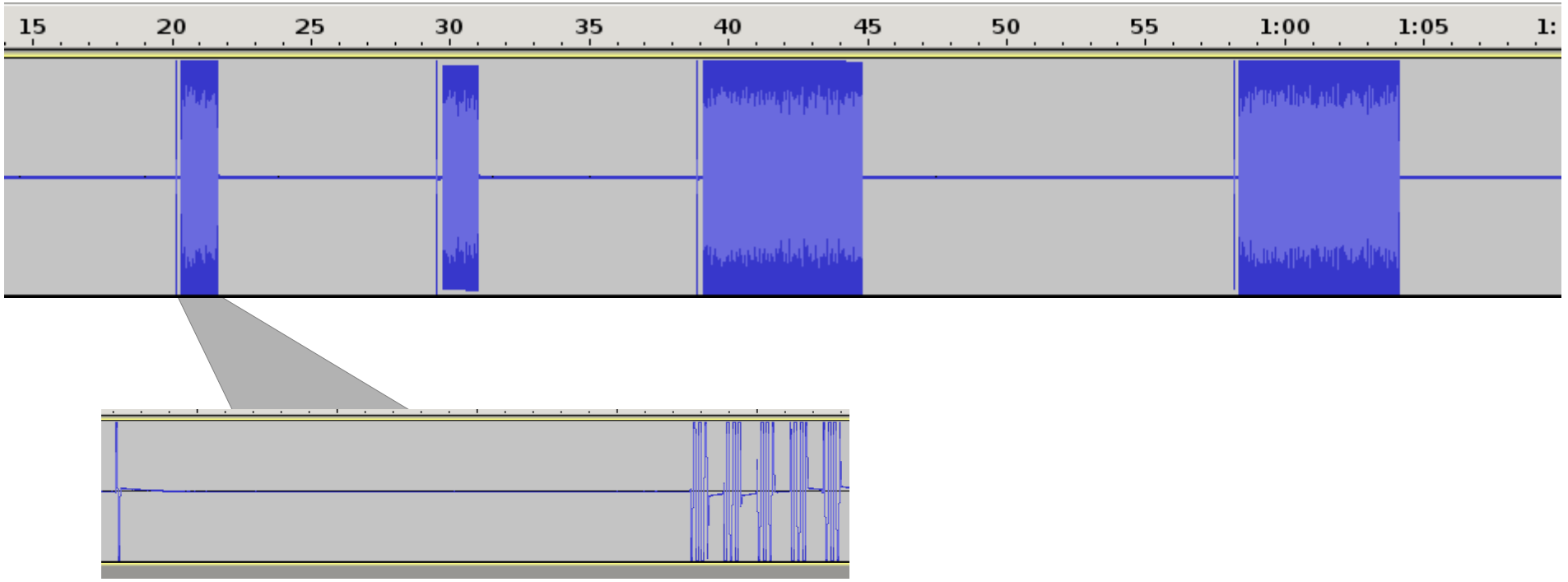
Identify the frequency

- **Slicing across different ranges: 310, 433, 868 and 915 MHz (e.g. RF explorer)**



Identify the modulation

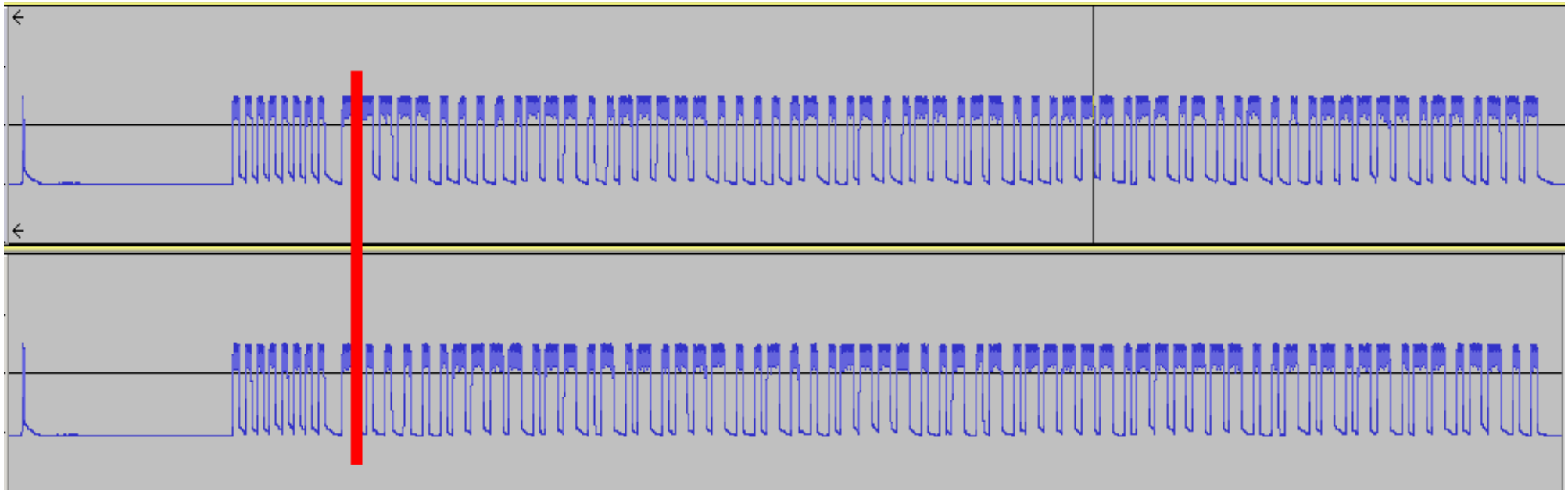
- **Capture 1 button (e.g with the HackRF)**



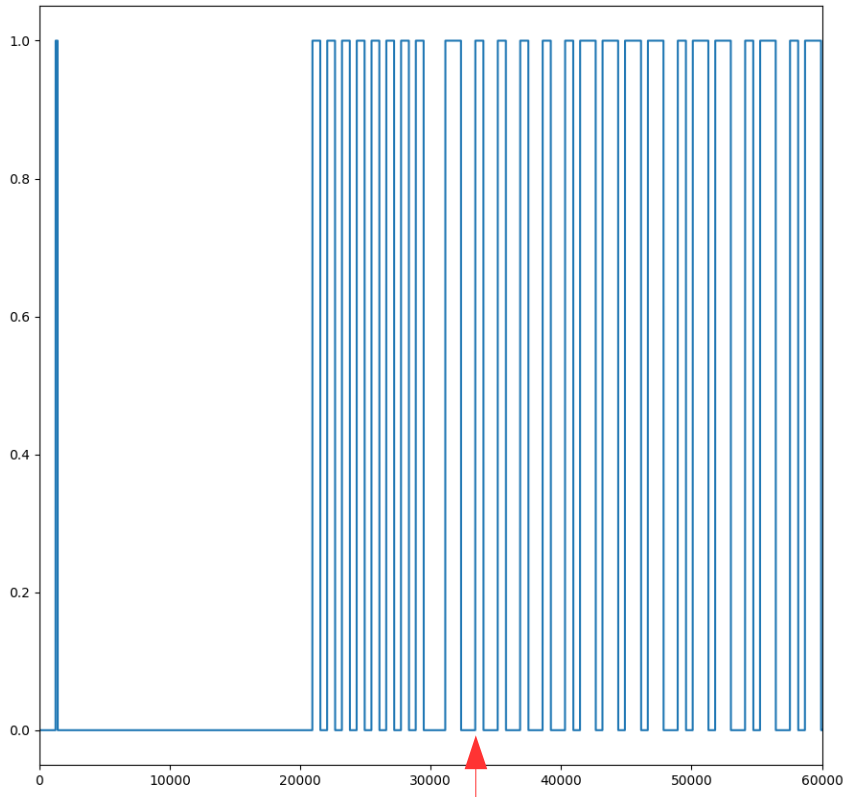
=> looks like a classic OOK modulation

Demodulation of the captured signal

- **We can perform an amplification demodulation (AM demod)**
- **But when comparing two pushes, the code is different:**

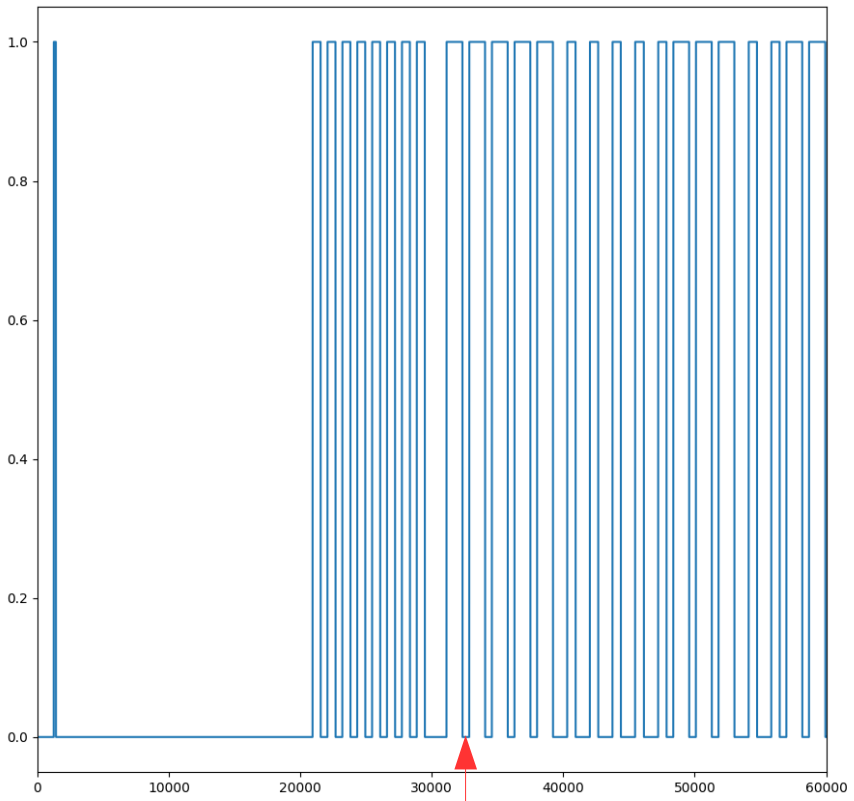


Cleaned tracks (e.g with scipy)



1st push

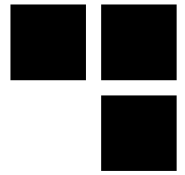
Short impulses



2nd push

Longer impulses here

Seems to be rolling code



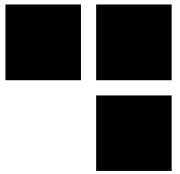
- **At each push the code is different → it's probably rolling code**
- **rolling code is briefly referenced when looking at product's documents**
- **replay is theoretically impossible**
- **but other attacks exist:**
  - side channels or firmware+memory dumping → recover the algorithm and the initial seed
  - cloning the next code → attack presented by Silvio Cesare at Black Hat USA 2014

To resume

- **The remote and Vigik are secure → almost good!**
  - Attacks on these devices exists but are kind tricky

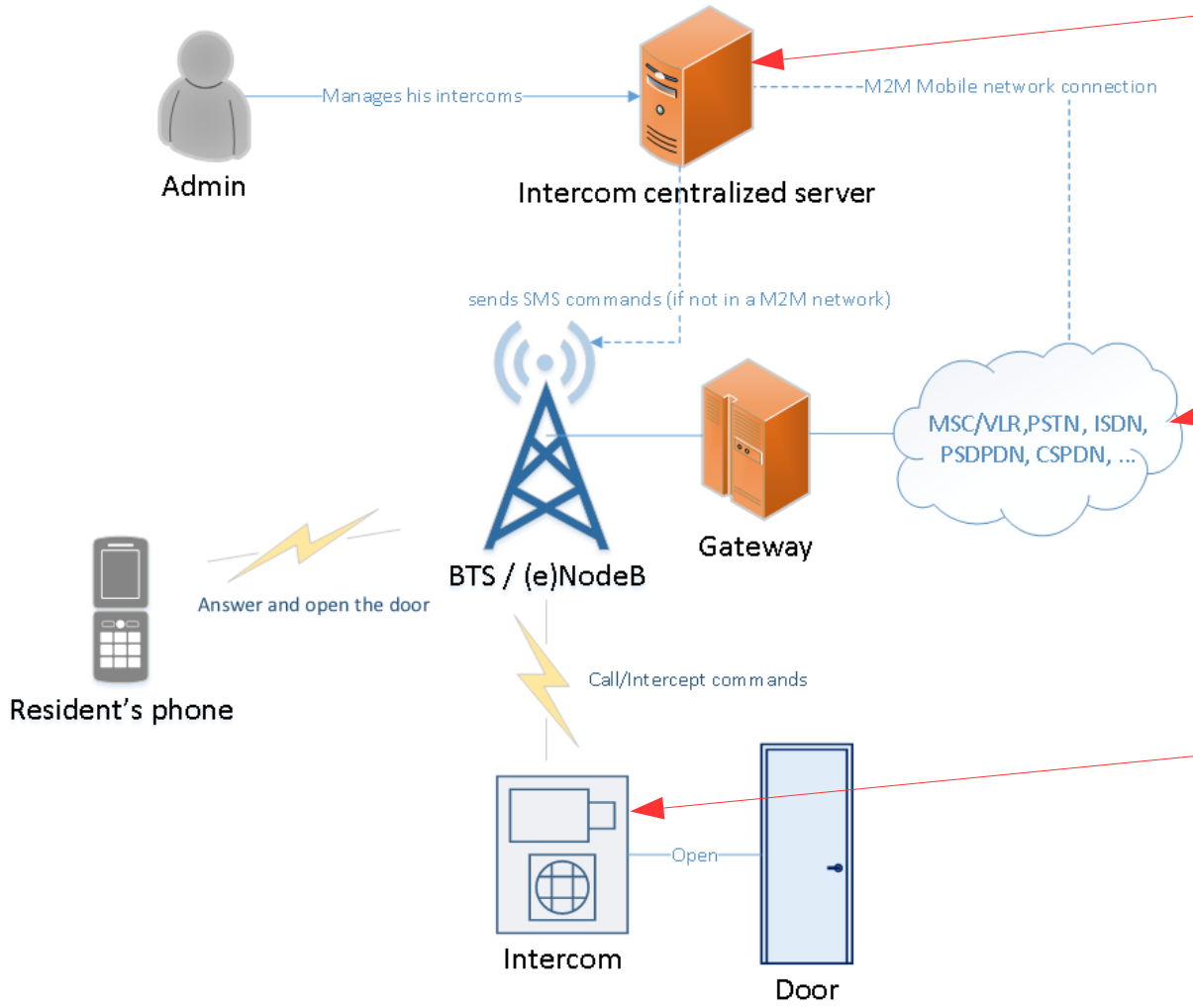
    → need the leak of Vigik token ID

    → or jam, relay 1 code and clone some rolling codes
- **These intercoms use 3G → could be downgraded to 2G → vulnerable to the previous attacks**
- **With 3G intercoms use TCP/IP stack → what about their web services?**



The mobile network approach

Attack vectors with M2M Intercoms

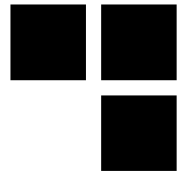


Vulnerabilities in Services: Web, SIP, etc.

SIM/USIM → look for vulnerabilities in the virtual network

3G downgrade to 2G + GSM interception

Website vulnerabilities



- **Websites → manage one or multiple intercoms thanks to their mobile number**
- **Vulnerabilities could be found:**
  - account guessing + bruteforce → we've tested it on a product
  - authentication bypasses → could be identified crawling with Google!
  - SQL injections,
  - LFI,
  - and so on.

Our tests on “Product A”

- **We’ve tested a 3G intercom that is provided with a M2M SIM Card**
- **Lets call it “Product A”**

Bruteforce accounts

- **By default, “Product A” website doesn’t enforce a password to manage intercoms:**

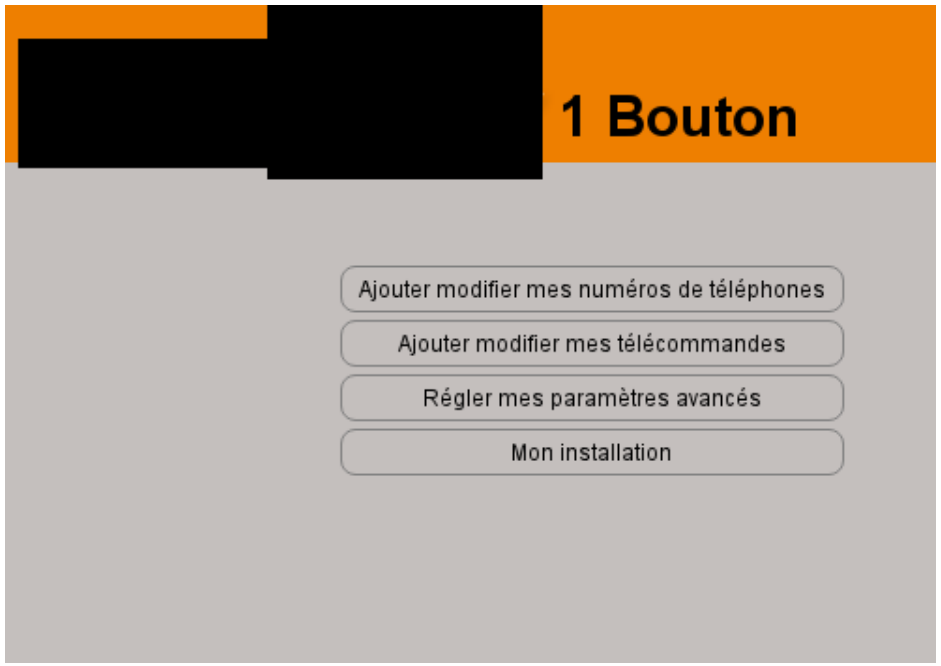
But we need a valid number…

Number enumeration (quick PoC)

```
url = "http://<login page of product A>/<login page>"
[...]
prefixes = [ "07", "08", "30", "70", "71", "72", "73", "74", […] ]
prefixes = reversed(sorted(prefixes))
init = 100000
numbers = []
for p in prefixes: # number generation
    init = 100000
    while init <= 999999:
        if init == 100000:
            numbers.append("06" + p + "000000")
        numbers.append("06" + p + str(init))
        Init += 1
f = open("numbers.list", "a+")
for x in numbers: # for each generated number, log existing account
    t = int(time.time()) # timestamp added for the POST query
    data = {"**CENSORED1**":x, "**CENSORED2**":t}
    r = requests.get(url, params=data, headers=headers)
    if r.url != u"http://<login page of product A>/<error page>":
        f.write(x+"\n")
```

Enumerated accounts

- **The server doesn’t mitigate wrong tries**
- **So 90 numbers have been enumerated for 1 prefix (+33 6 77 ******) < 4 hours**
- **We are able to manage intercoms without the need of SDR tools!**



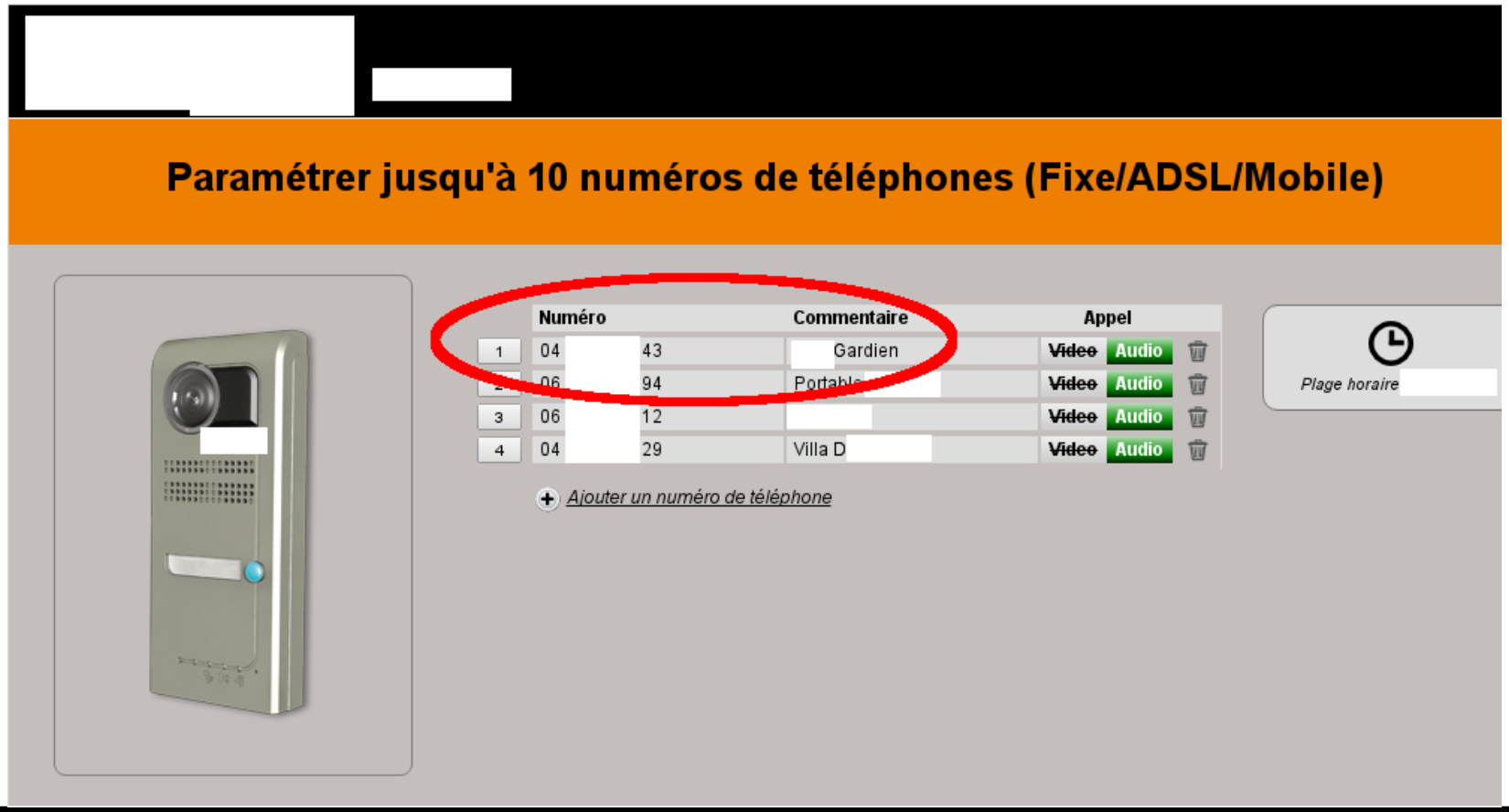
Attack scenarios

- **Without the need of any SDR tool:**
  - Update all intercoms with a premium rate number =/

  - open doors → but we need the locations...

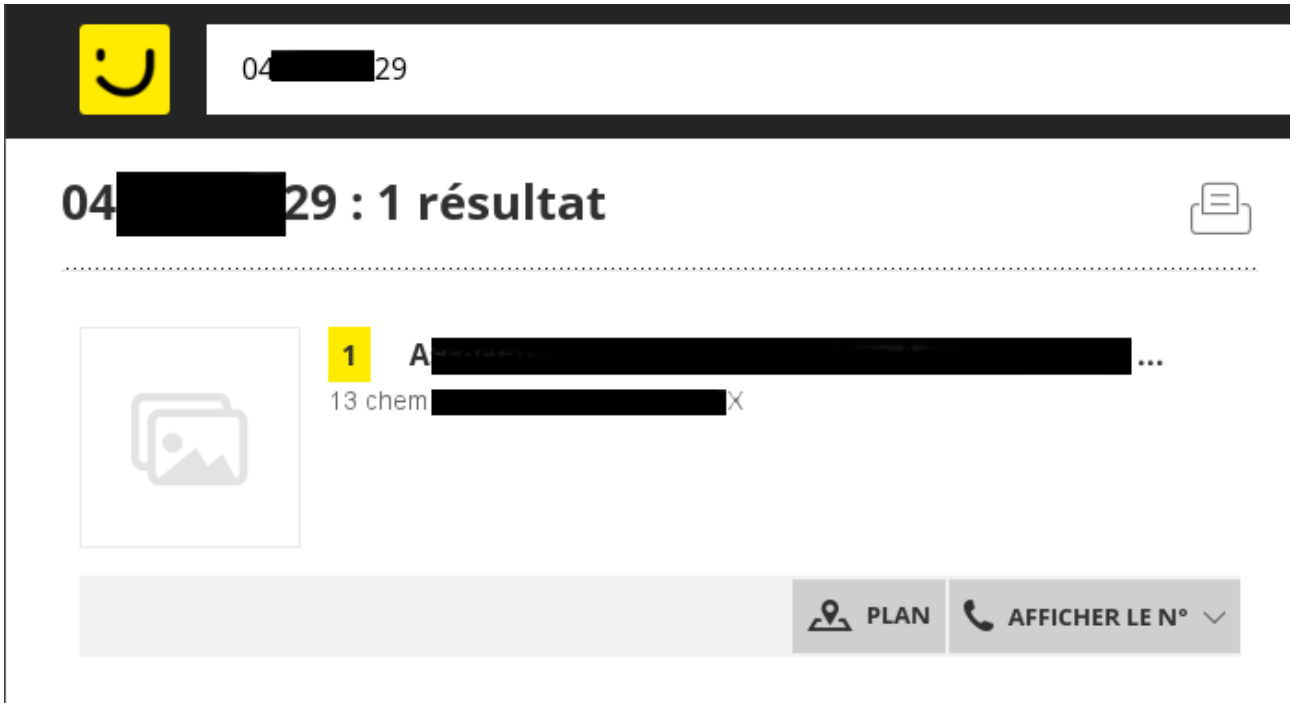
How to get the location?

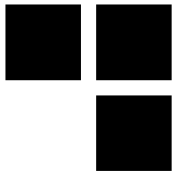
- **In general, people add their home number first...**



Reverse look-up directories

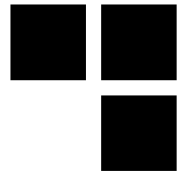
- **Reverse look-up directories → get the precise location**





Get a free internet connection or intrude Product A's private network

The M2M virtual network as a second attack vector



- **Provided SIM/USIM cards could be plugged on other devices**

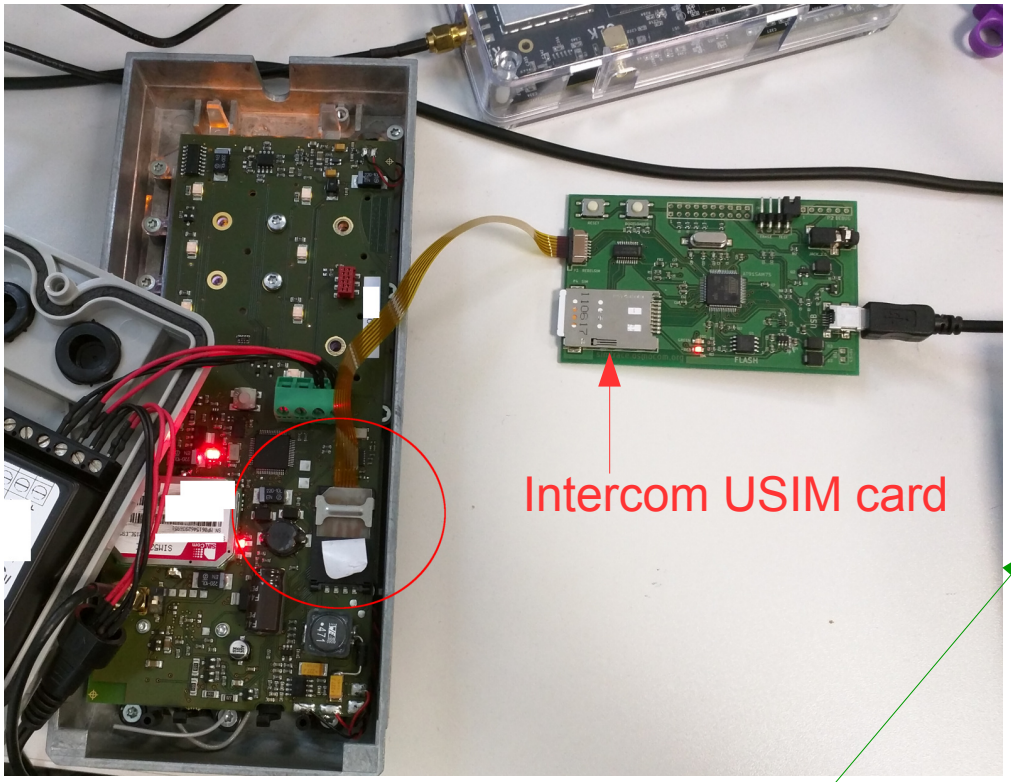
  → we can scan the virtual network

- **But product's "A" SIM/USIM card has a PIN code… =/**

  → not a problem for the SIMtrace tool!

SIMtrace setup and results

SIMtrace as a “proxy” between the SIM/USIM ↔ intercom:



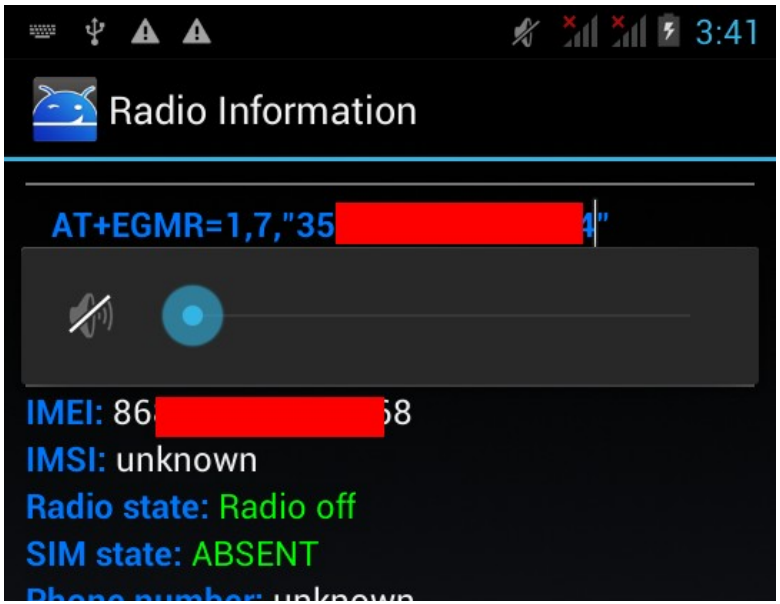

```
Entering main loop
ATR APDU: 3b 9f 96 80 1f c7 80 31
e0 73 fe 21 1b 64 40 91 11 00 82
90 00 01
PPS(Fi=9/Di=6) APDU: 00 a4 00 04
02 3f 00 61 23
[…]
APDU: 00 20 00 01 08 ** ** ** ** ff
ff ff ff 90 00
APDU: 00 2c 00 01 00 63 ca
[…]
```

PIN code typed by the “Product A” intercom itself

Connecting to the M2M network

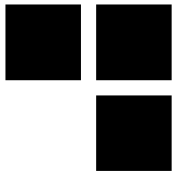
- **Put the SIM/USIM in your phone**
- **Optionally change the IMEI (possible with some Chinese phones)**
- **Setup the right APN (Access Point Name) of the M2M network → documented**
- **Tether the communication**

  → use a computer



Changing the IMEI within the engineer mode

Traceroute in the M2M virtual network

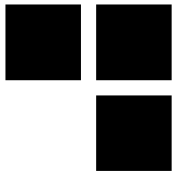


Check the connection with a tethered computer:

```
$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  192.168.42.129 (192.168.42.129)  0.622 ms  0.643 ms  0.705 ms
 2  10.***.***.250 (10.***.***.250)  105.629 ms  125.547 ms  185.628 ms
 3  10.***.***.209 (10.***.***.209)  195.783 ms  195.900 ms  195.831 ms
[...]
14  google-public-dns-a.google.com (8.8.8.8)  50.771 ms  50.248 ms
51.016 ms
```

An attacker will now be able to:

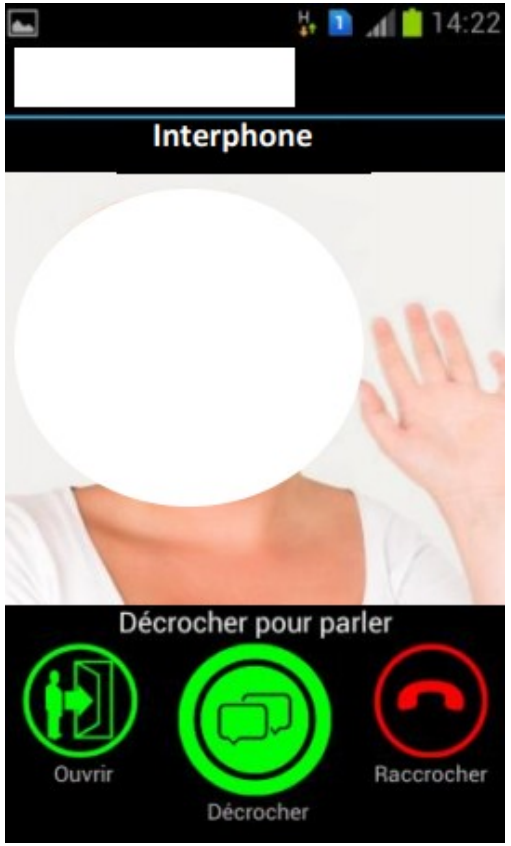
1) scan the virtual network
2) search for vulnerable services
3) then exploit vulnerable services
4) and so on… or use the SIM/USIM to get a free internet access ^^



SIP as another attack vector

SIP as an attack surface

- **“Product A” has a mobile application to provide Video calls**
- **Video calls use SIP**
- **To use this app a premium account is required =(**
- **But let’s analyze it!**

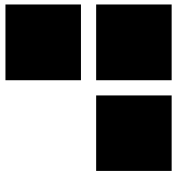


Application analysis: first results

- **Mainly (very) bad/NULL SSL checks… → MITM possible**
- **Also one SIP credential seems to be hardcoded:**

```
public static final String [redacted]_PASSWD = "EK[redacted]D";
public static final String [redacted]_SENDER_ID = "97[redacted]7";
public static final String [redacted]_SETTINGS = "[redacted].settings";
public static final String [redacted]_URL = "https://sip.[redacted]/";
public static final String [redacted]_URL_TEST = "https://siptest.[redacted]/";
public static final String [redacted]_USER = "user";
public static final String [redacted]_USERNAME_INTERPHONE_SIP = "1002";
```

Registering in the SIP server



- **Using hardcoded credentials → success!**

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP
10.***.***.11:38703;alias;branch=z9hG4bK.rfZ5uXs1W;rpor
t=38703;received=19**********2
From: <sip:user@sip.**********>;tag=qmu7Mgc8t
To: sip:user@sip.**********;tag=**********
CSeq: 21 REGISTER
```

Results on “product A” SIP vector

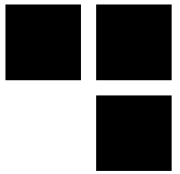
- **Not satisfying =/**
- **We are able to contact simple users like:**
  - “user”;
  - “root”, …
- **But impossible to contact a known number**

  → Maybe because the number needs be registered as a premium extension
- **Sandro Gauci (@sandrogauci) helped to understand my issue → we need a valid prefix format and/or a valid device connected to that platform to begin to go further =/**
- **This vector as been set in stand by → waiting to get an access to the premium offer**

Security recommendations for M2M solutions

- **Enforce a PIN code on SIM/USIM cards → like in “Product A”**
- **Whitelist IMEIs**
- **Audit/pentest regularly the management website against web vulnerabilities, but also other services**
- **Restrict actions and requests on APNs**
- **Firewall the virtual network, or do some segmentation**
- **Audit/pentest the virtual network against network attacks and vulnerabilities in services**
- **Monitor and block SIM/USIM cards that have a suspicious behavior**

Conclusion



- **With GSM intercoms we can:**
  - open a door
  - call premium rate numbers
  - spy on conversations if ATS0 is supported
- **Intercoms using the mobile network → same flaws as mobile phones**
- **Other devices in the IoT ecosystem use the mobile network**
- **M2M intercoms introduce new vectors of attack → much more destructive → require a simple Internet connection (no SDR tools needed)**
  - But M2M SIM/USIM cards are also used in many other IoT products!

ANY QUESTIONS?

Thanks for your attention !

SYNACKTIV
DIGITAL SECURITY