Intercoms Hacking

Call the Front Door and Install Your Back Door

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!**
- **Do red team tests at Synacktiv:**
  - spear phishing,
  - remote and

    physical intrusions
  - ...

Physical intrusions (1)

- **Why?:**
  - to plug a malicious device,
  - dump computer memory,
  - or let malicious USB keys indoor, ...

Physical intrusion (2)

- **Main problem: we always need a way to enter to a building**
- **How?:**
  - lockpicking,
  - RF attacks,
  - social engineering,
  - or attacking Intercoms!

Red team tests

- **Sometimes it works, but sometimes we get spotted...**

Alert! Intruder!

Why intercoms could be interesting?

- **At night → entering premises like a ninja!**
- **But also:**
  - to spy on conversations in the street, when it’s possible
  - to make money
  - and have a lot of fun...

Warning

- **This talk applies practical attacks on intercoms**
- **But other devices in the “IoT” ecosystem are also concerned...**

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

* Like +49<cell phone number> in Germany

Human curiosity...

- **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!

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

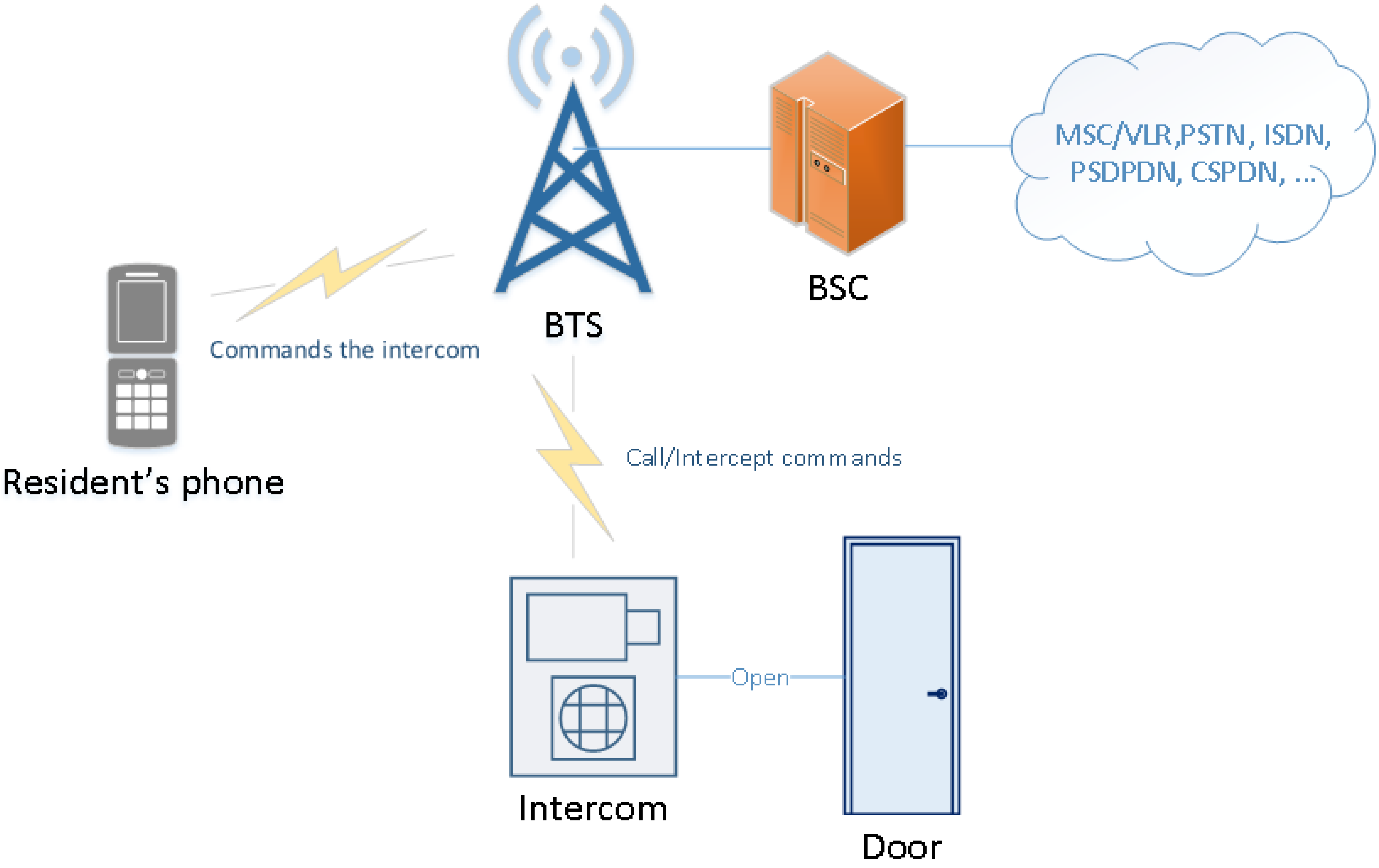
Numeric intercoms

Wires replaced by:

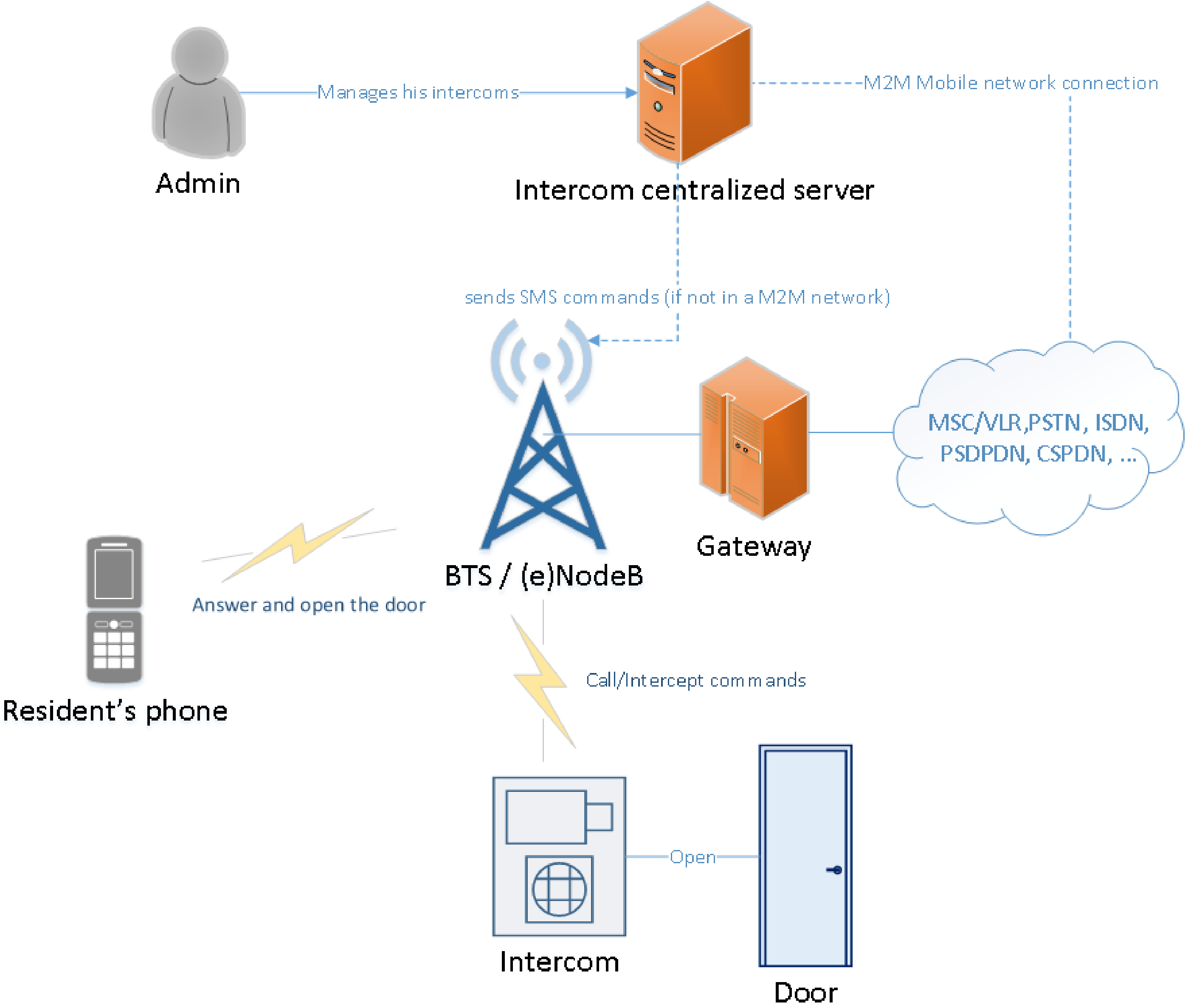
- GSM, 3G, rarely in 4G
- or Wi-Fi…

⇒ Avoid complicated and cumbersome cables
⇒ Easy installation

Numeric intercoms: simplified architecture



Network architecture with M2M


Admin
Manages his intercoms
Intercom centralized server
M2M Mobile network connection
sends SMS commands (if not in a M2M network)
MSC/VLR,PSTN, ISDN, PSDPDN, CSPDN, ...
Gateway
BTS / (e)NodeB
Answer and open the door
Resident's phone
Call/Intercept commands
Open
Intercom
Door

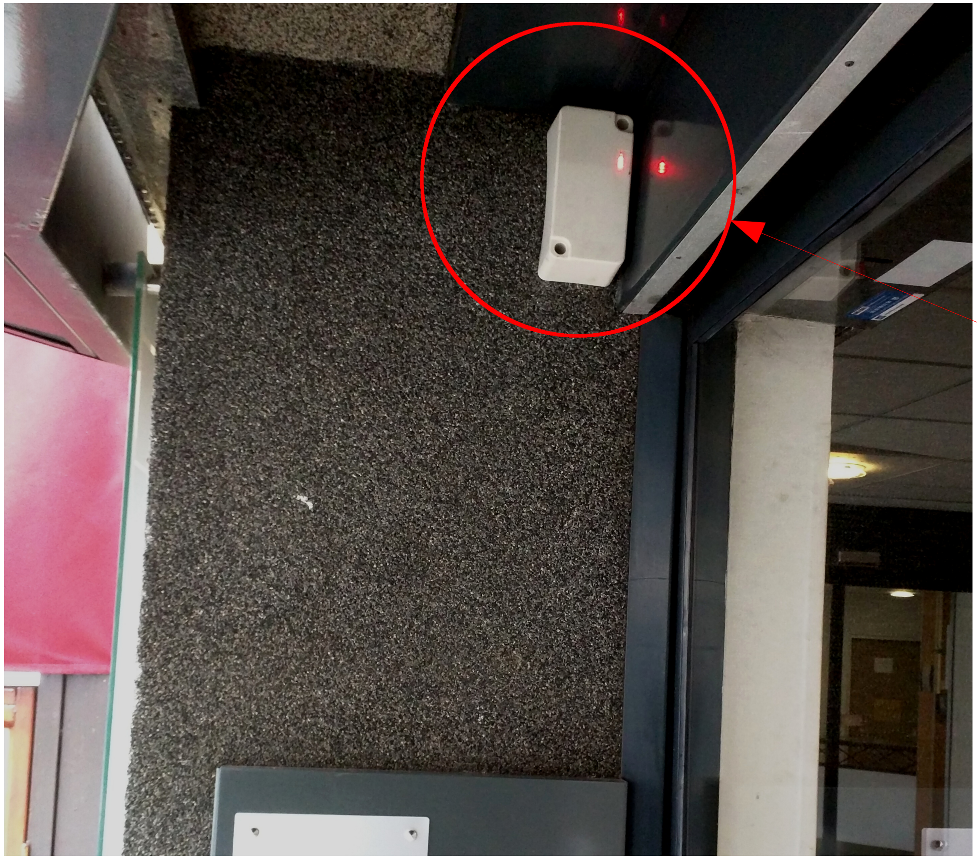
Different brands market

- **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**

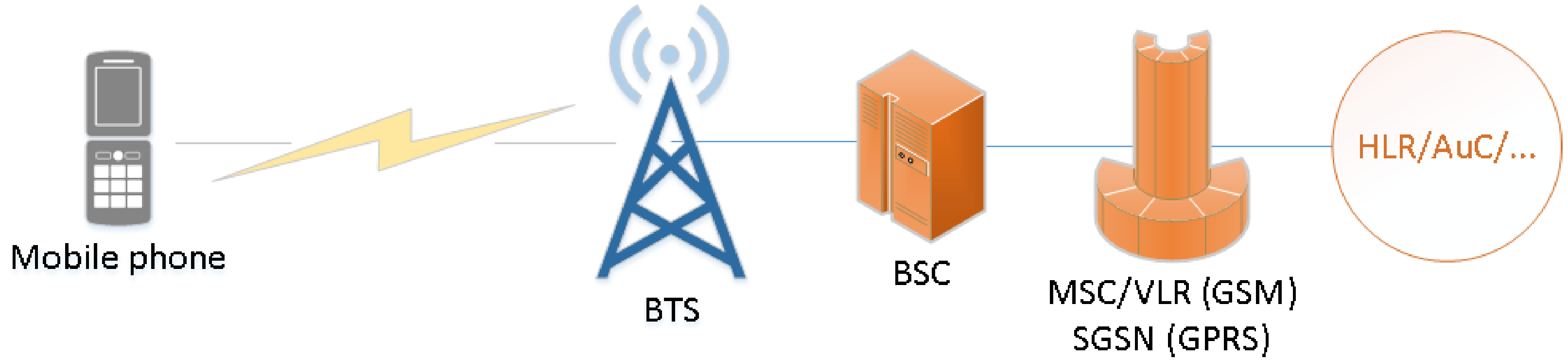
    (In October 2015, Altaf Shaik, Ravishankar Borgaonkar, N. Asokan, Valtteri Niemi and Jean-Pierre Seifert)
  - **How to not break LTE crypto**

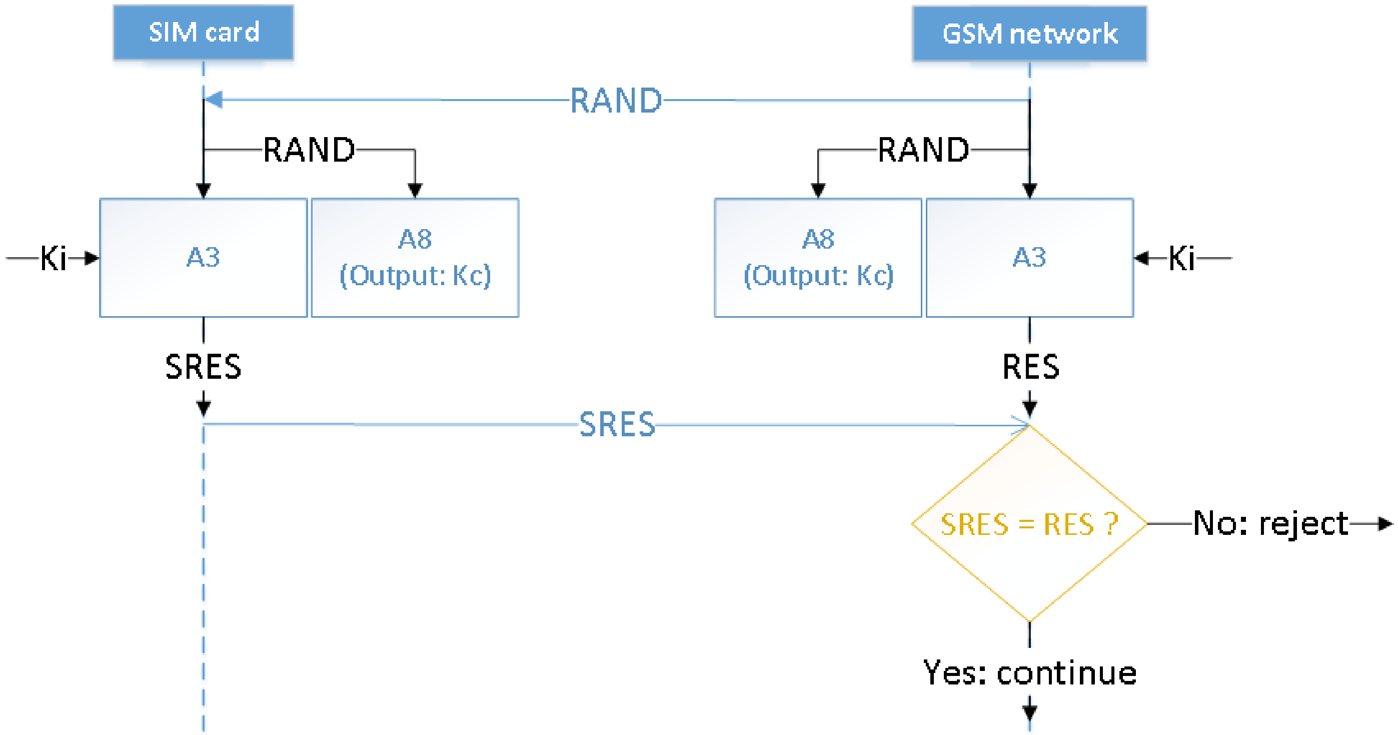
    (at SSTIC 2016, Christophe Devine and Benoit Michau)
  - And many others...

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

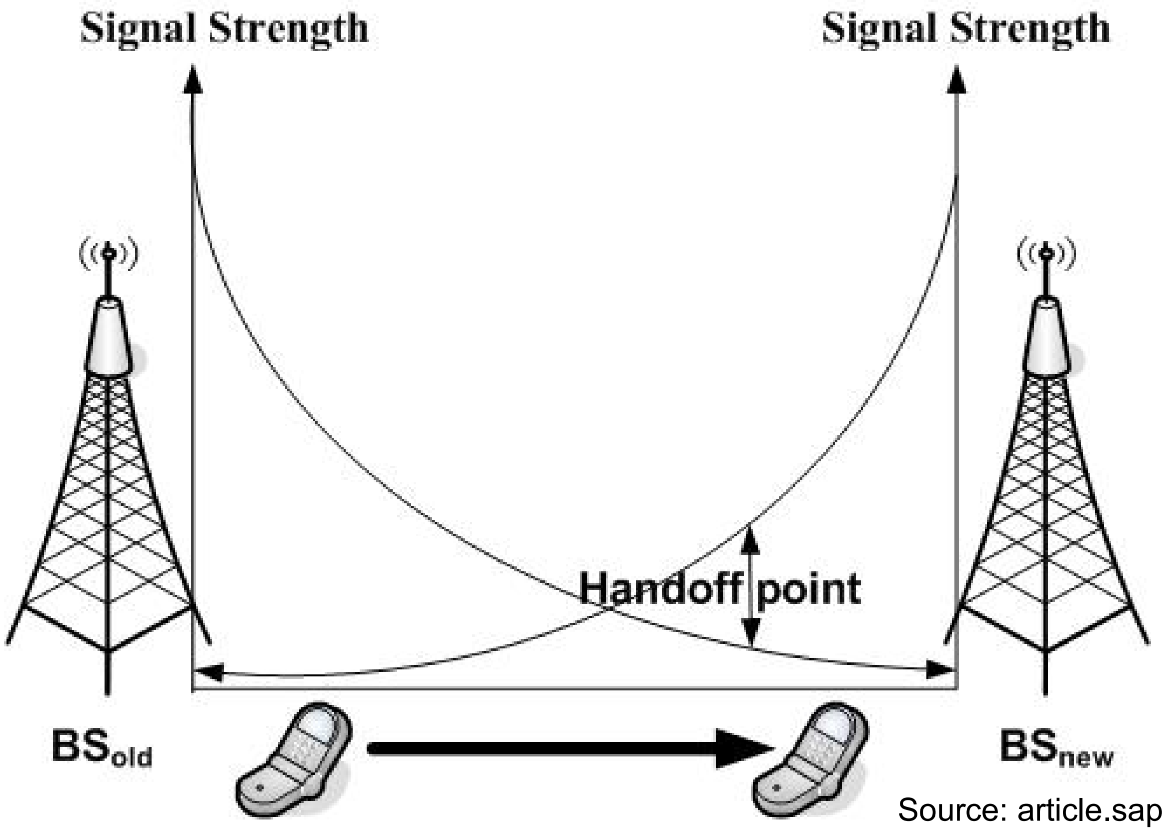
GSM and GPRS: authentication





- 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

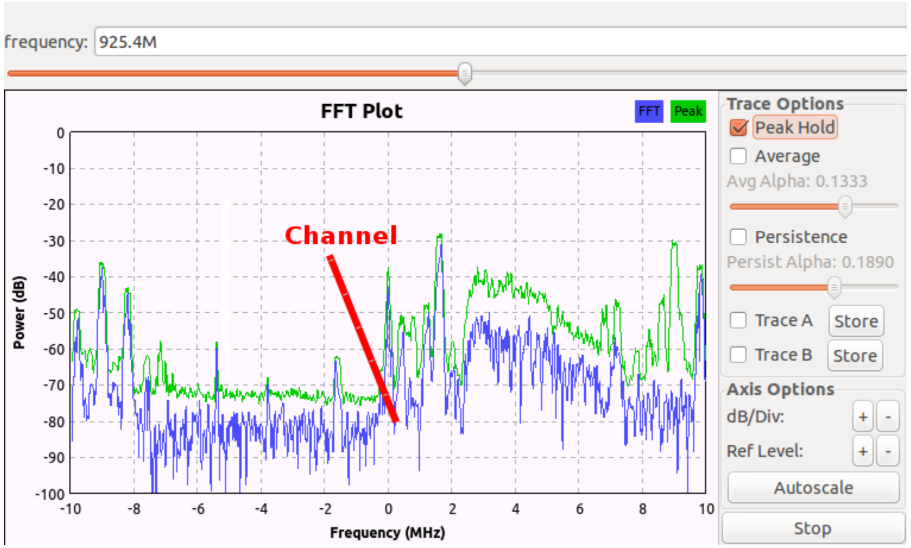
	GSM	3G	4G
Client authentication	YES	YES	YES
Network authentication	NO	Only if USIM is used (not SIM)	YES
Signaling integrity	NO	YES	YES
Encryption	A5/1	KASUMI | SNOW-3G	SNOW-3G | AES | ZUC...

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...



frequency: 925.4M

☑ Peak Hold

□ Average

Avg Alpha: 0.1333

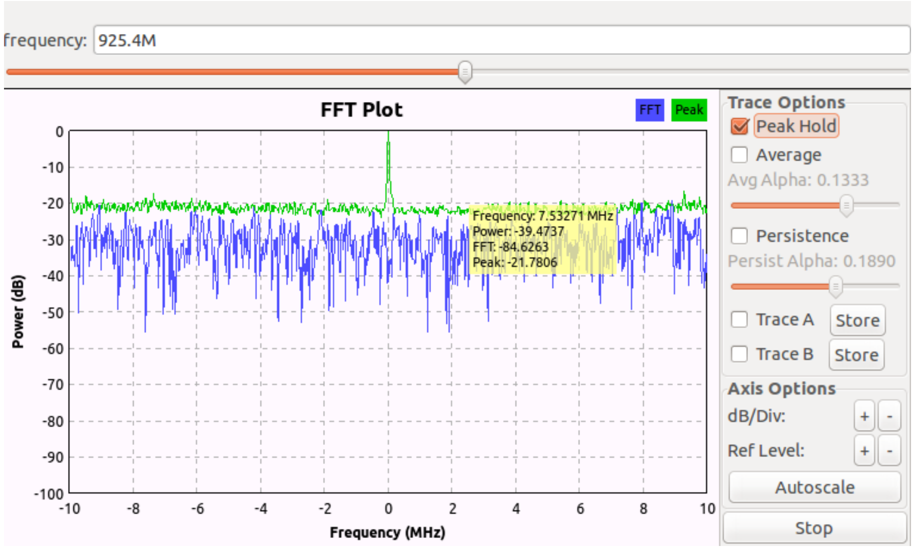
□ Persistence

Persist Alpha: 0.1890

□ Trace A

□ Trace B

Before



frequency: 925.4M

Frequency: 7.53271 MHz
Power: -39.4737
FFT: -84.6263
Peak: -21.7806

☑ Peak Hold

□ Average

Avg Alpha: 0.1333

□ Persistence

Persist Alpha: 0.1890

□ Trace A

□ Trace B

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. »

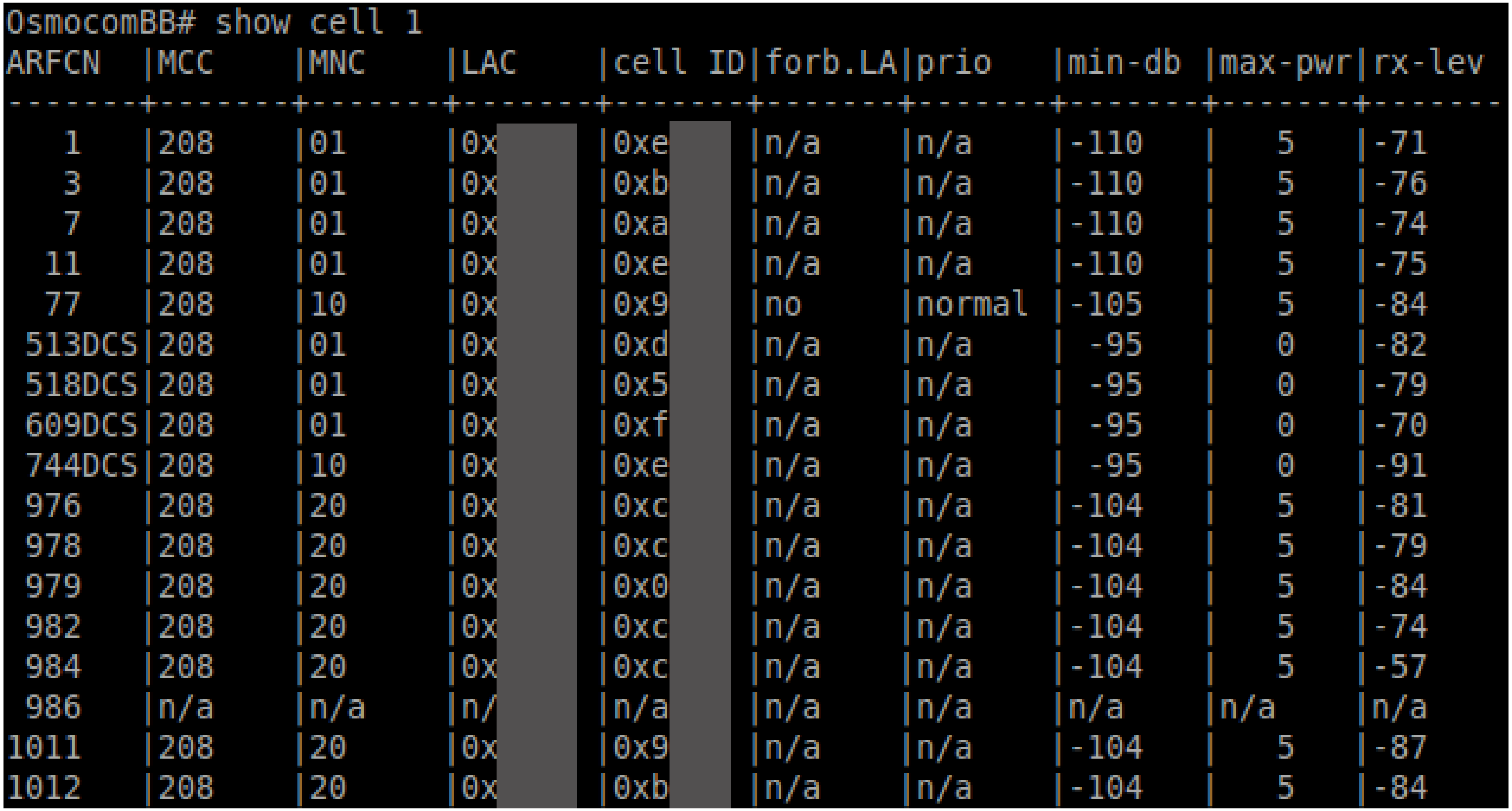
= If 3G is unreachable → use 2G instead!

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)

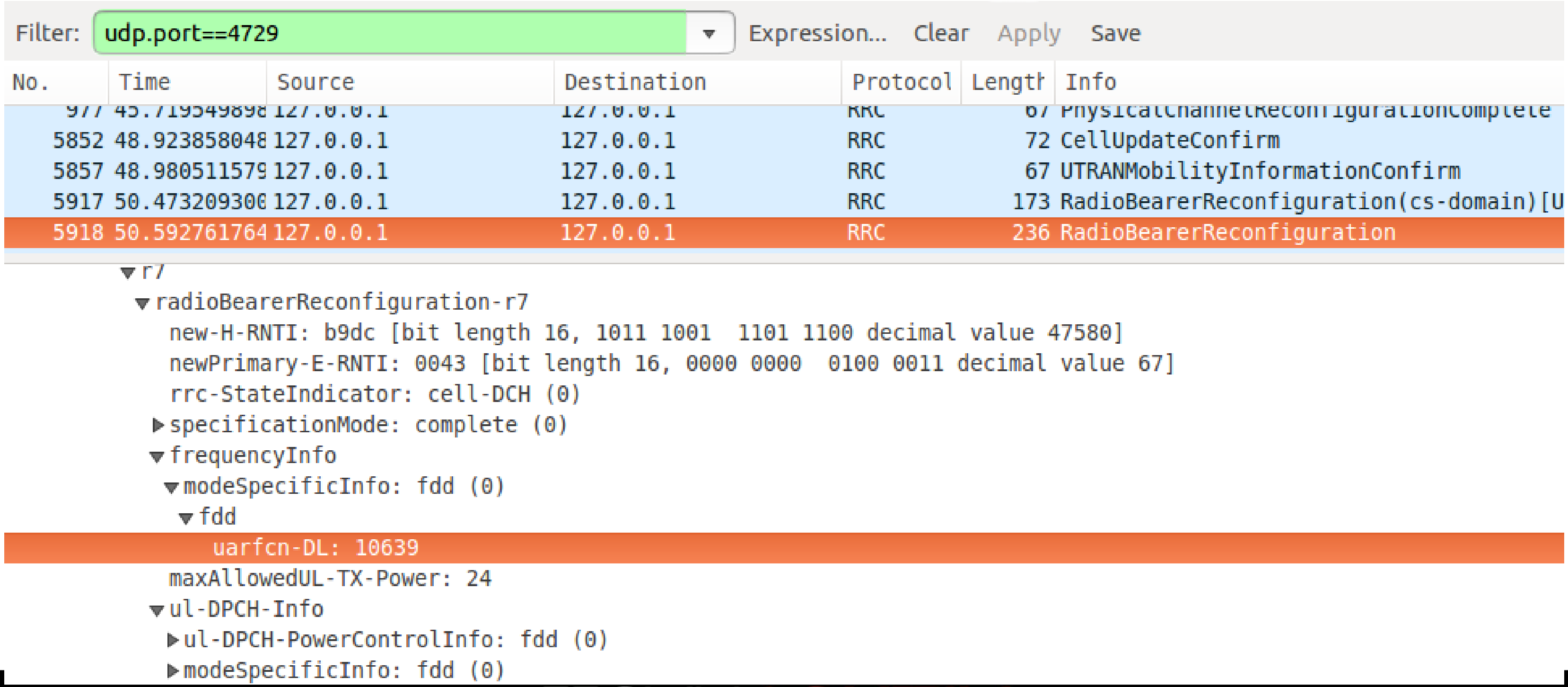
- **OsmocomBB only works for GSM =(**



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



```
Filter: udp.port==4729
```

```
No.    Time            Source        Destination   Protocol  Length  Info
  977  45.719549898    127.0.0.1     127.0.0.1     RRC         67    PhysicalChannelReconfigurationComplete
 5852  48.923858048    127.0.0.1     127.0.0.1     RRC         72    CellUpdateConfirm
 5857  48.980511579    127.0.0.1     127.0.0.1     RRC         67    UTRANMobilityInformationConfirm
 5917  50.473209306    127.0.0.1     127.0.0.1     RRC        173    RadioBearerReconfiguration(cs-domain)[U
 5918  50.592761764    127.0.0.1     127.0.0.1     RRC        236    RadioBearerReconfiguration
```

```
        ▼r7
         ▼radioBearerReconfiguration-r7
            new-H-RNTI: b9dc [bit length 16, 1011 1001  1101 1100 decimal value 47580]
            newPrimary-E-RNTI: 0043 [bit length 16, 0000 0000  0100 0011 decimal value 67]
            rrc-StateIndicator: cell-DCH (0)
          ▶specificationMode: complete (0)
          ▼frequencyInfo
            ▼modeSpecificInfo: fdd (0)
              ▼fdd
                  uarfcn-DL: 10639
            maxAllowedUL-TX-Power: 24
          ▼ul-DPCH-Info
            ▶ul-DPCH-PowerControlInfo: fdd (0)
            ▶modeSpecificInfo: fdd (0)
```

Baseband diag interfaces (2)

- **Qualcomm baseband 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<<
[...]
```

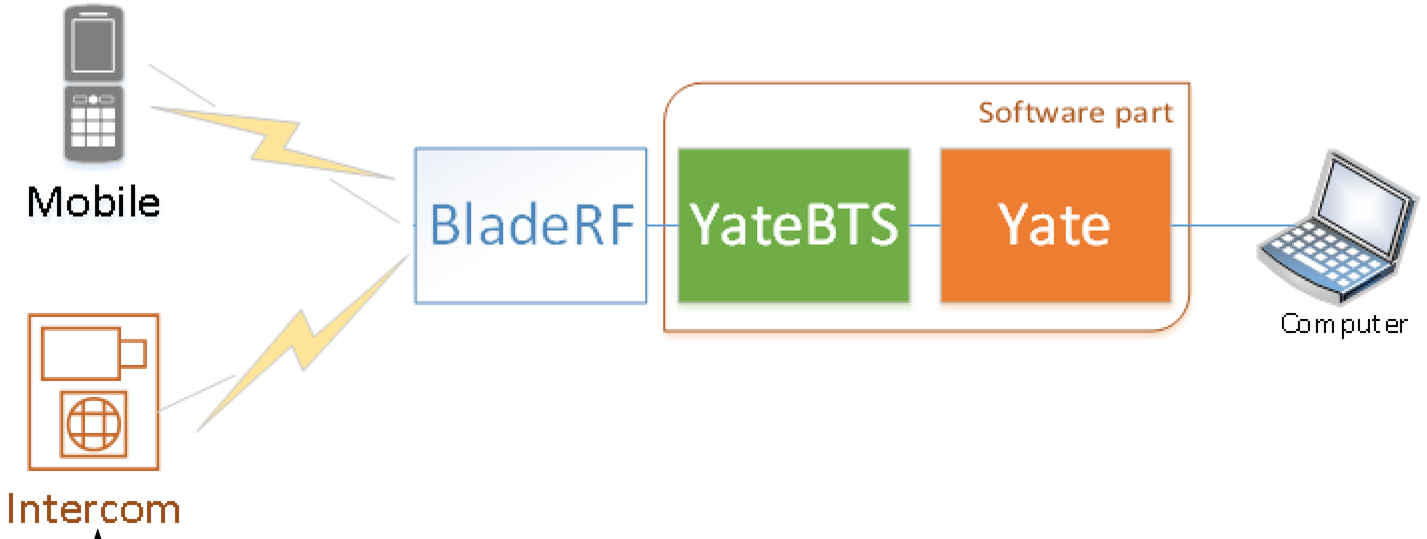
Downgrade 3G → 2G demo

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

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

GSM Lab setup: for interception

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



Linkcom iDP GSM

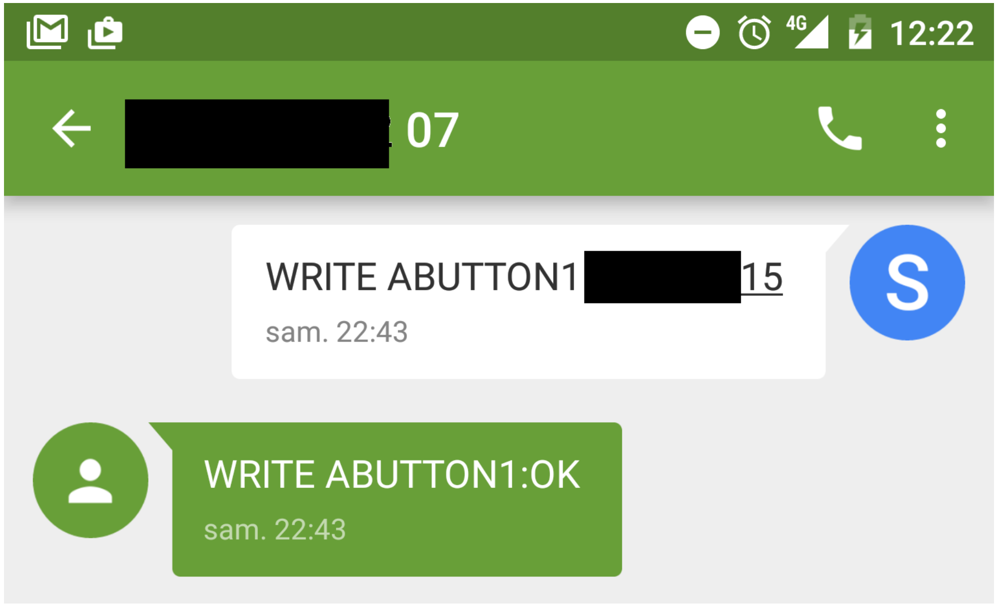
- 1 BladeRF = 370 € minimum
- 2 Antennas = 15 € minimum each
- YateBTS software = FREE
- **Total cost = 400 €**

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**



WRITE ABUTTON1[redacted]15
sam. 22:43

WRITE ABUTTON1:OK
sam. 22:43

Hypotheses as a potential attacker

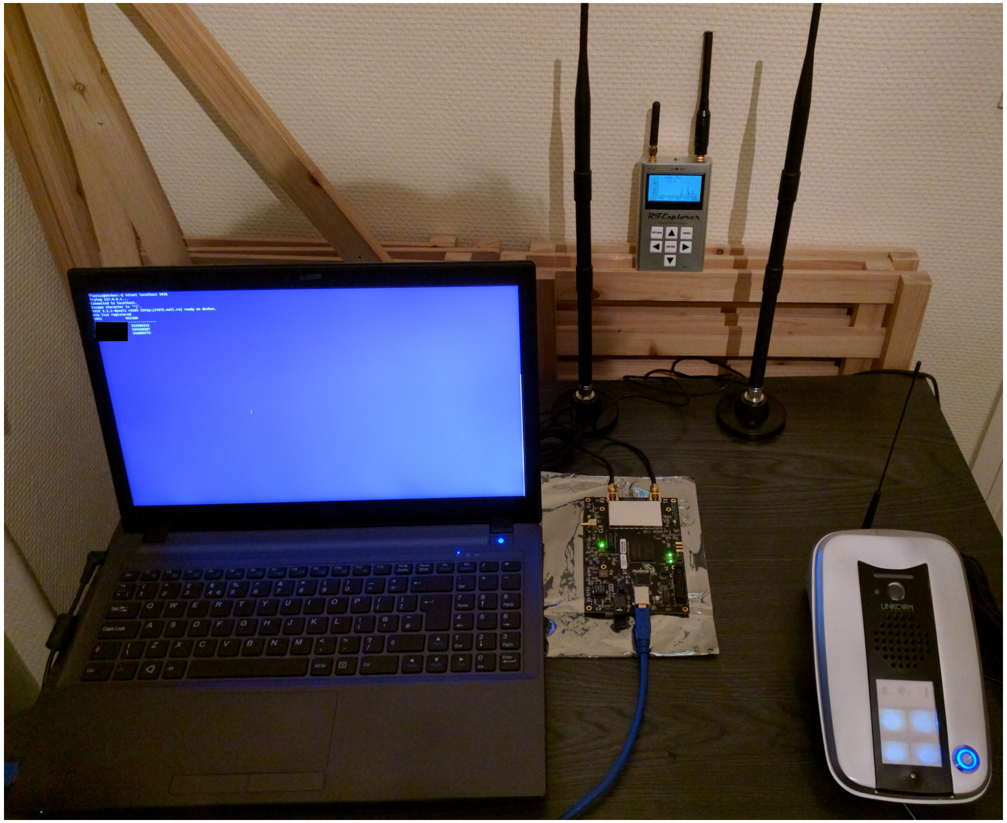
- **We don’t know the mobile operator**
- **We don’t know intercom’s number**
- **The commands could be found with public or leaked documentations, or by performing a firmware analysis**

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

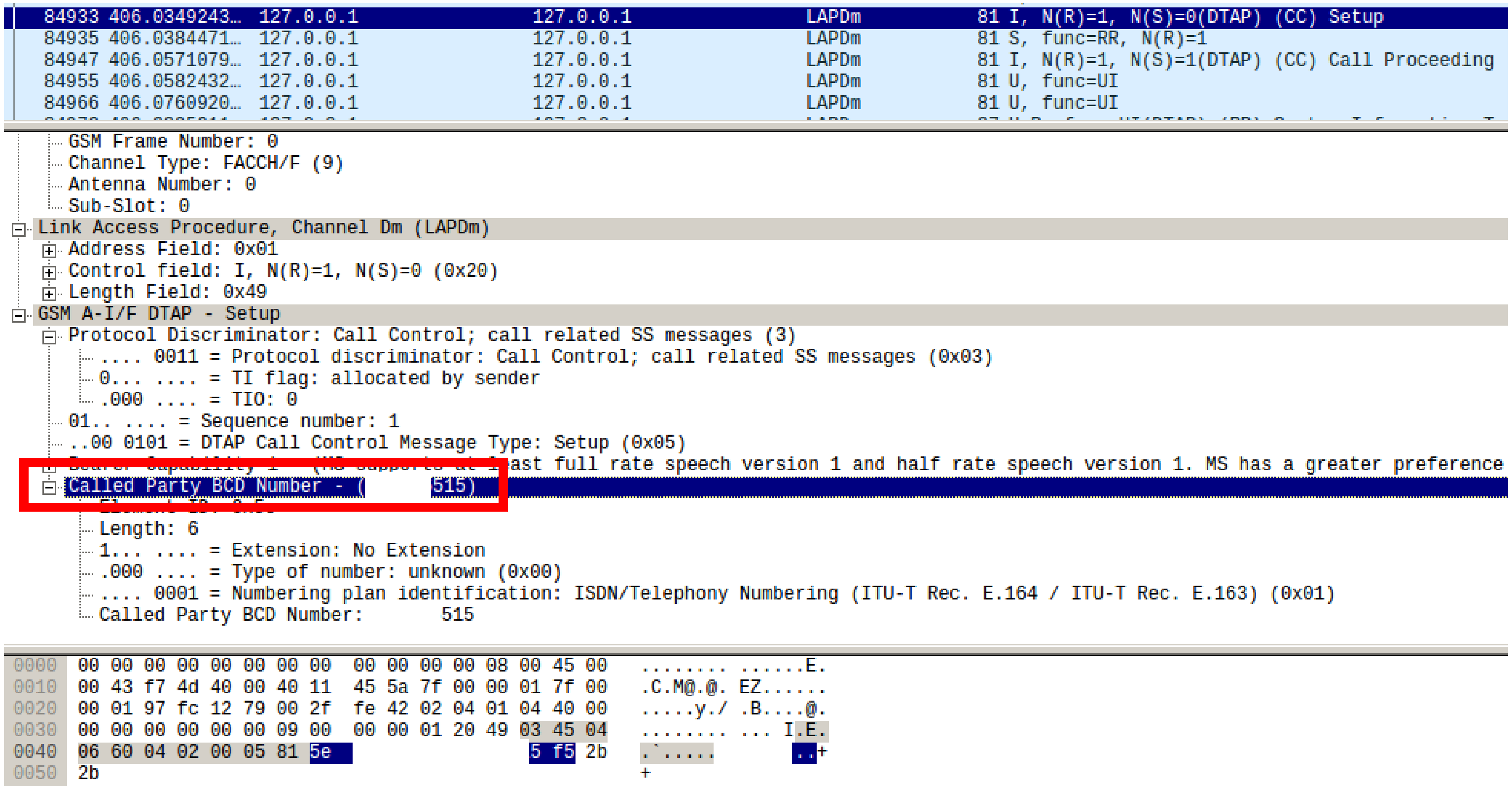
- **Bruteforcing the 4 MCC/MNC (FR)**
  - 15min~ waiting for each MCC/MNC
- **Strong GSM signal**
- **Button push → calling intercepted → success!**



<u>Note</u>: 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)
  ⊞ Bearer Capability 1 - (MS supports at least full rate speech version 1 and half rate speech version 1. MS has a greater preference
  ⊟ Called Party BCD Number - ([redacted]515)
      Element ID: 0x5e
      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: [redacted]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 [redacted] 5 f5 2b   .`..... [redacted] ..+
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?

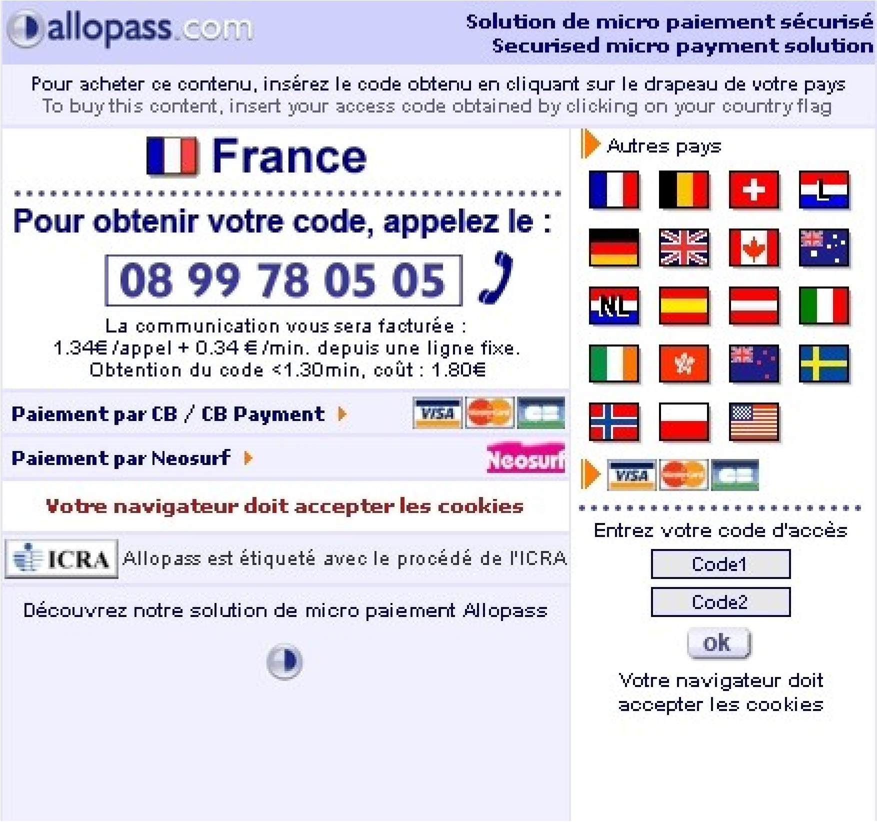
- **We can interact with Intercom’s baseband:**
  - retrieve SMS messages → *AT+CMGL="ALL"*
  - spying building door conversations with auto-answer feature (if not disabled) → *ATS0=1*
  - and so on.

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.



allopass.com

Solution de micro paiement sécurisé
Securised micro payment solution

Pour acheter ce contenu, insérez le code obtenu en cliquant sur le drapeau de votre pays
To buy this content, insert your access code obtained by clicking on your country flag

France

Pour obtenir votre code, appelez le :

08 99 78 05 05

La communication vous sera facturée :
1.34€ /appel + 0.34 € /min. depuis une ligne fixe.
Obtention du code <1.30min, coût : 1.80€

Paiement par CB / CB Payment

Paiement par Neosurf

Votre navigateur doit accepter les cookies

ICRA Allopass est étiqueté avec le procédé de l'ICRA

Découvrez notre solution de micro paiement Allopass

Autres pays

Entrez votre code d'accès

Votre navigateur doit accepter les cookies

Attacking 3G/4G intercoms

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

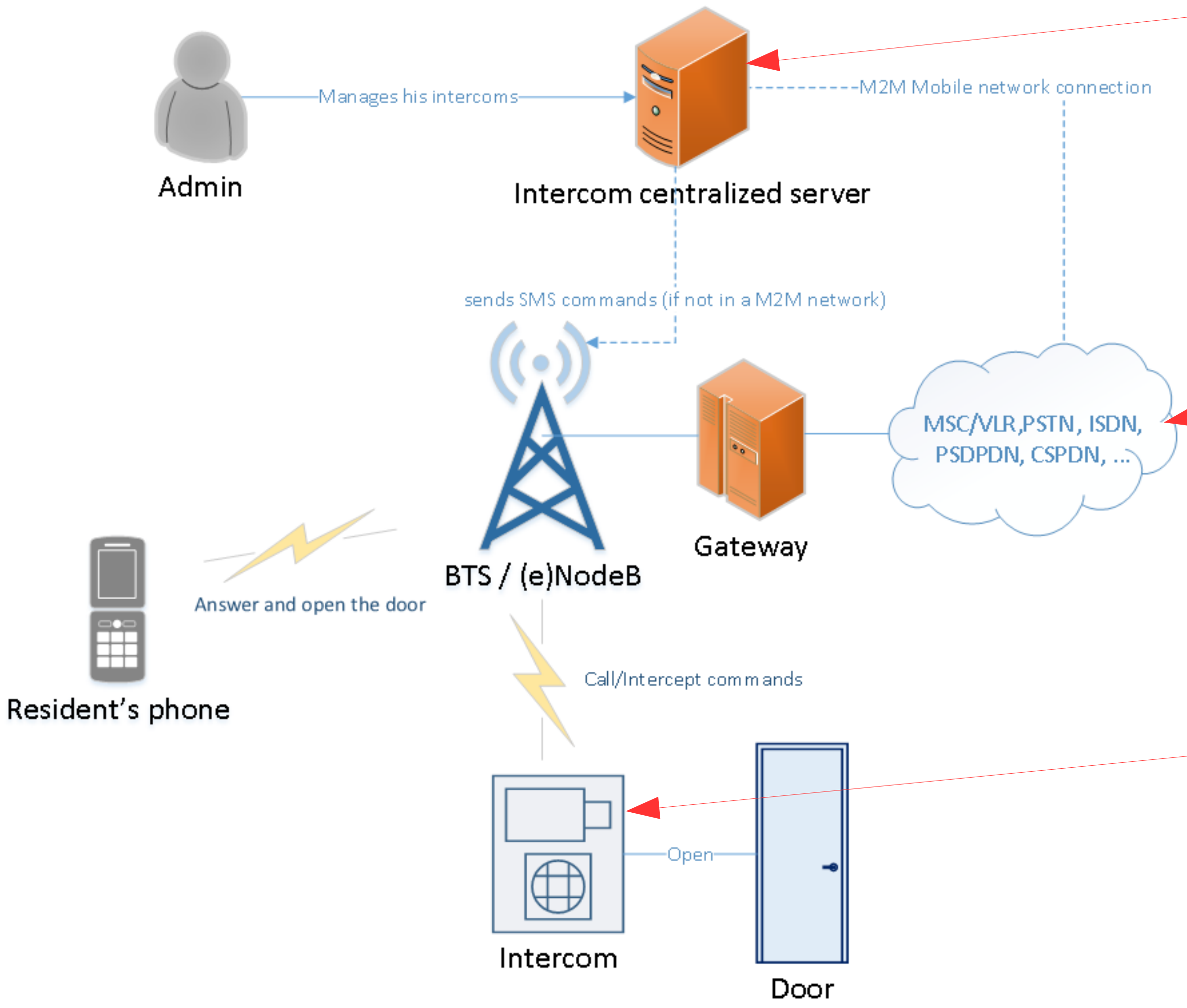
- **Use the UMTS network by default**

  → 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...

Attack vectors with M2M Intercoms


Admin
Manages his intercoms
Intercom centralized server
M2M Mobile network connection
sends SMS commands (if not in a M2M network)
MSC/VLR,PSTN, ISDN, PSDPDN, CSPDN, ...
Gateway
BTS / (e)NodeB
Answer and open the door
Resident's phone
Call/Intercept commands
Open
Intercom
Door

Vulnerabilities in Services: Web, SIP, etc.

SIM/USIM → look for vulnerabilities in the virtual network

3G downgrade to 2G + GSM interception

Website vulnerabilities

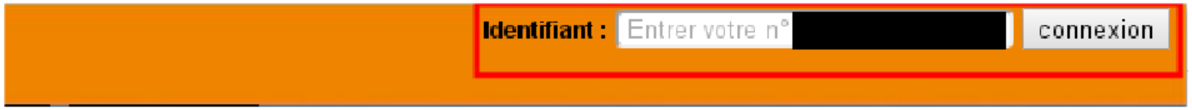
- **Websites → manage one or multiple intercoms thanks to their mobile number**
- **Vulnerabilities could be found:**
  - account guessing + bruteforce → we've tested it on a product
  - authentication bypasses → could be identified crawling with Google!
  - SQL injections,
  - LFI,
  - and so on.

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:**



Identifiant : Entrer votre n°[redacted]

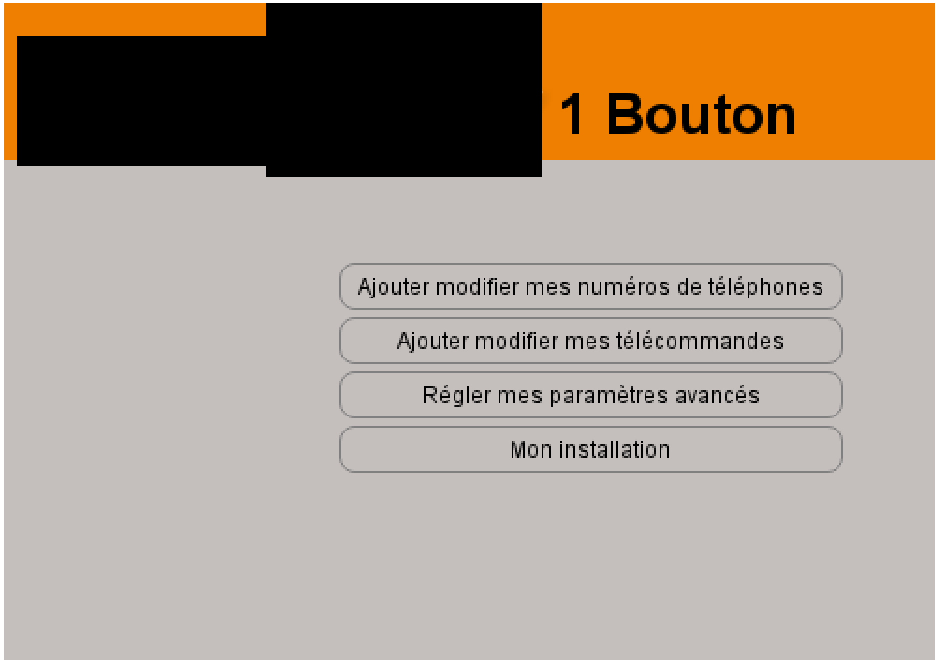
But we need a valid number…

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!



[redacted] 1 Bouton

Ajouter modifier mes numéros de téléphones
Ajouter modifier mes télécommandes
Régler mes paramètres avancés
Mon installation

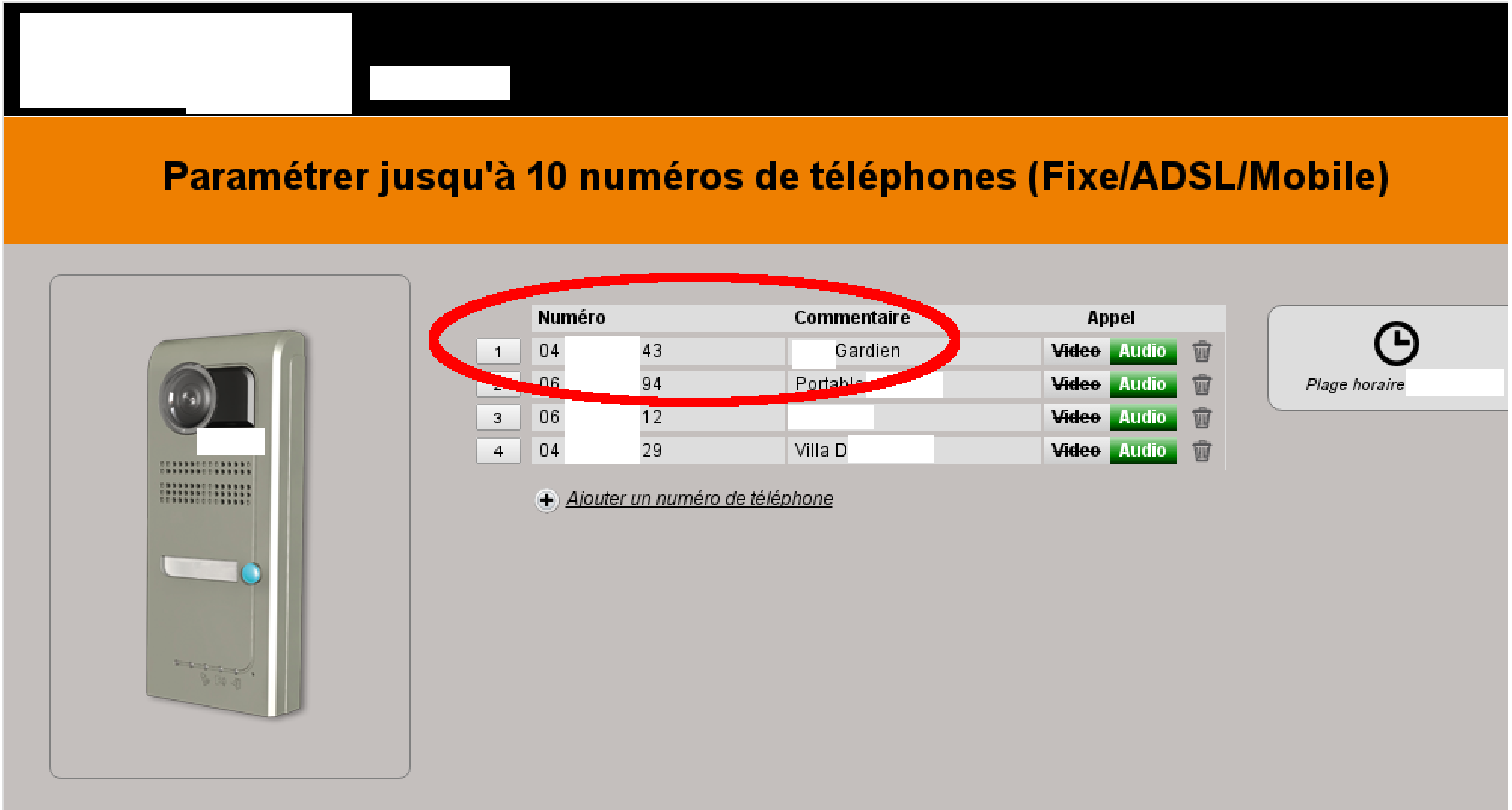
Attack scenarios

- **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...**



[redacted] [redacted]

Paramétrer jusqu'à 10 numéros de téléphones (Fixe/ADSL/Mobile)

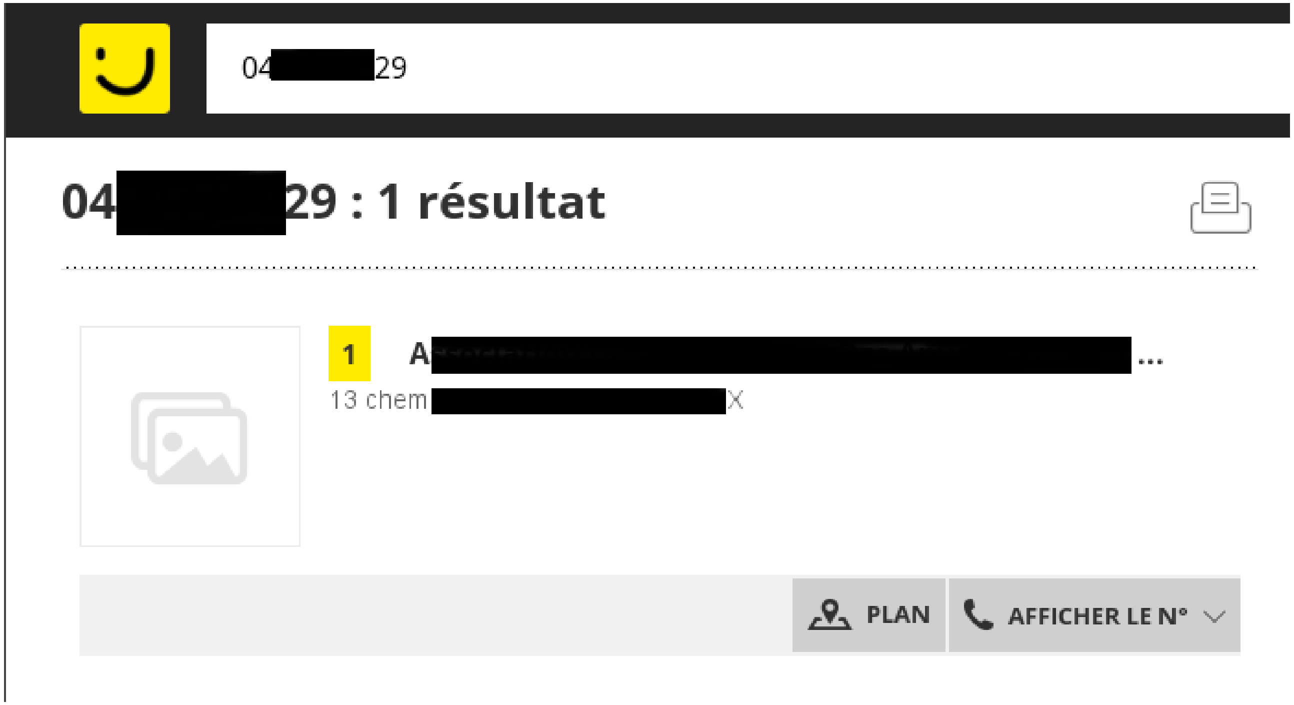
	Numéro	Commentaire	Appel	
1	04 [redacted] 43	[redacted] Gardien	~~Video~~ Audio	🗑
2	06 [redacted] 94	Portable [redacted]	~~Video~~ Audio	🗑
3	06 [redacted] 12	[redacted]	~~Video~~ Audio	🗑
4	04 [redacted] 29	Villa D [redacted]	~~Video~~ Audio	🗑

⊕ *Ajouter un numéro de téléphone*

Plage horaire [redacted]

Reverse look-up directories

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



04[redacted]29

04[redacted]29 : 1 résultat

1 **A[redacted] ...**

13 chem[redacted]X

The M2M virtual network as a second attack vector

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

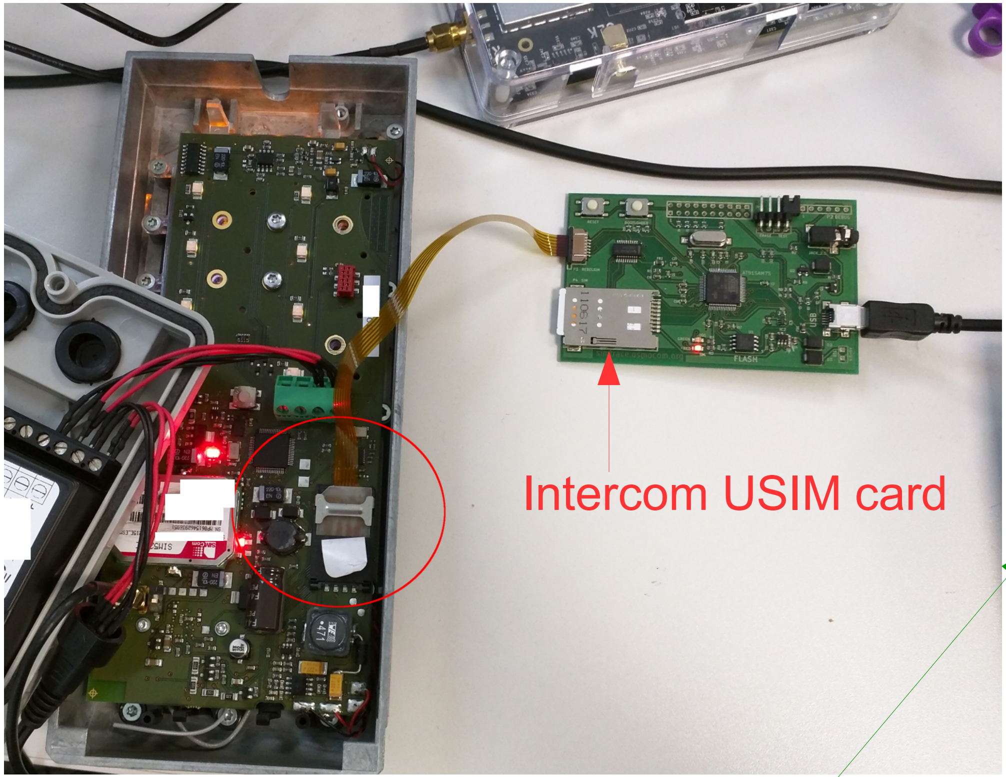
  → we can scan the virtual network

- **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:



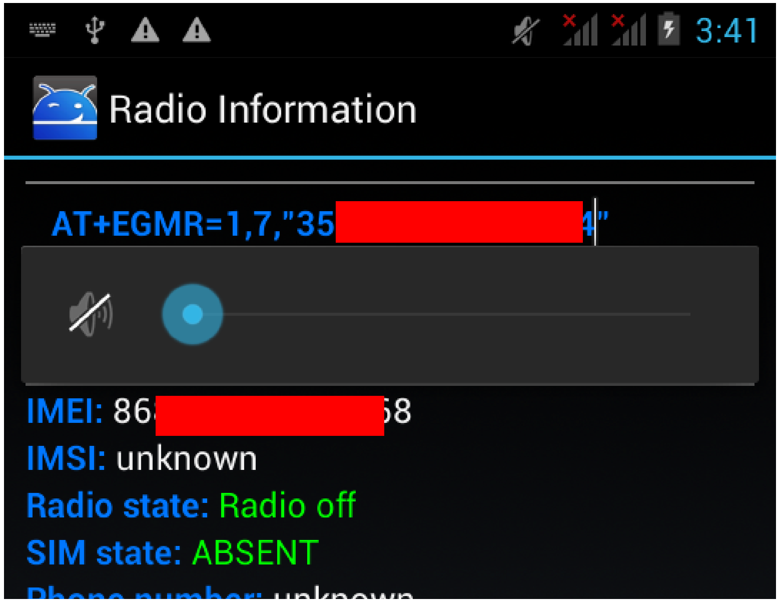
```
Entering main loop
ATR APDU: 3b 9f 96 80 1f c7 80 31
e0 73 fe 21 1b 64 40 91 11 00 82
90 00 01
PPS(Fi=9/Di=6) APDU: 00 a4 00 04
02 3f 00 61 23
[…]
APDU: 00 20 00 01 08 ** ** ** ** ff
ff ff ff 90 00
APDU: 00 2c 00 01 00 63 ca
[…]
```

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 <u>right APN</u> (Access Point Name) of the M2M network → documented**
- **Tether the communication**

  → use a computer



Radio Information

AT+EGMR=1,7,"35[redacted]"

IMEI: 86[redacted]8
IMSI: unknown
Radio state: Radio off
SIM state: ABSENT
Phone number: unknown

Changing the IMEI within the engineer mode

Traceroute in the M2M virtual network

Check the connection with a tethered computer:

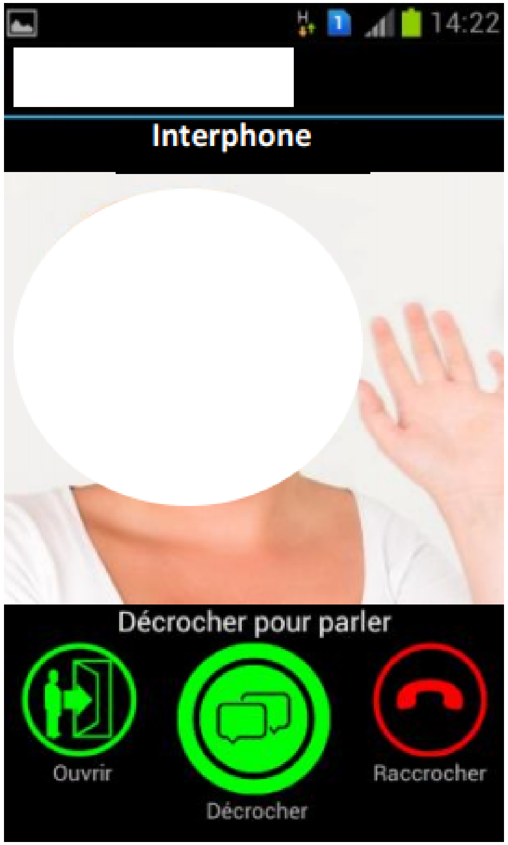
```
$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  192.168.42.129 (192.168.42.129)  0.622 ms  0.643 ms  0.705 ms
 2  10.***.***.250 (10.***.***.250)  105.629 ms  125.547 ms  185.628 ms
 3  10.***.***.209 (10.***.***.209)  195.783 ms  195.900 ms  195.831 ms
[...]
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 ^^

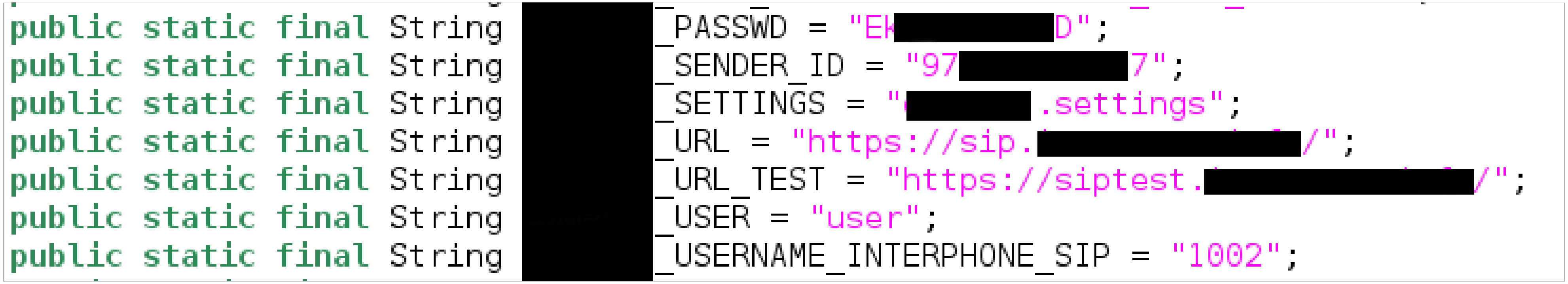
SIP as an attack surface

- "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 is possible**
- **Also one SIP credential seems to be hardcoded:**



```
public static final String [redacted]_PASSWD = "E[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
- **Actual question: how to find a valid extension (without flooding with “INVITE” requests)?**

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!
- **Further work:**
  - find a solution about the SIP vector,
  - start attacking intercoms' basebands,
  - reduce the lab with an odroid device or another alternative :)

ANY QUESTIONS?

Thanks for your attention !

SYNACKTIV
DIGITAL SECURITY