Advanced Security for Systems Engineering – VO 08: Mobile Applications

Clemens Hlauschek
Christian Brem
**Agenda**

**GSM Security**
- GSM/UMTS/LTE Introduction
- Attack Overview
- IMSI Catcher Internals

**Android Security**
- Reversing Android Applications
- Hooking and Instrumentation
GSM Security
Secret US Spy Program

(See Wall Street Journal – Nov 2014)
IMSI Catcher: Identify Protesters

These are fake mobile base stations, whose only purpose is electronic surveillance and tracking of people’s mobile phones, nearby.

(See https://gitlab.com/Hounge/Android-IMSI-Catcher-Detector)
IMSI Catcher / Stingray

- Researchers found 18 IMSI Catcher in Washington D.C. within 2 days

Attacks:

- Location Tracking
- Call / SMS eavesdropping
- MitM against data link
- SMS injection
IMSI Catcher: Low Cost

- Attacks can be launched by anyone nowadays
- Huge security and privacy problem!
- Starting 1000 EUR for HW-Equipment
- OpenSource projects:
  - Osmocon OpenBSC
  - OpenBTS
  - OpenLTE
  - srsLTE
IMSI Catcher: More Attacks

Deliver Spam:

- IMSI Catcher concealed in car, drive through city
- Spammers injected 6000 messages in half an hour
- Charged 1.000 Yuan (142 EUR) per 1000 users

- Attack vulnerable UICC / Baseband firmware / ... 
- Reconfigure phone – permanent MitM via Access Point Name (APN) change
- Intercept 2-factor auth (mTan)
History of 3GPP Networks and Main Security Issues

- **2G/GSM since 1991, GPRS**
  - Location privacy
  - No mutual authentication
  - Weak encryption: A5/1, A5/2

- **3G/UMTS since 2001**
  - Location privacy
  - Mutual authentication / strong encryption but
  - Downgrade to 2G often possible
History of 3GPP Networks and Main Security Issues

- **4G/LTE**, deployment started: 2009
  - Security problems of 3G mostly not solved
  - Mainly performance improvements

- **5G**, deployment started: 2019
  - Better privacy (encrypted SUPI/IMSI)
3GPP Networks: Main Security Issues

- 2G backward compatibility will remain for some time
- Devices always connect to base station with strongest signal
- Base station decides protocol version / encryption
- Core Network (Switching, SS7): No authentication
  - Query encryption key (2G, 3G)
  - Inject spoofed SMS
  - Reroute and eavesdrop on calls
  - Track subscribers worldwide
- Large-Scale DoS attacks
  - Race condition: Paging requests
- Femtocells: Cheap MitM attacks for 3g/4g
Structure of a GSM network

ME: Mobile Equipment
MT/TE

SIM
UICC

BTS: Base Transceiver Station

GERAN: GSM EDGE Radio Access Network
BSS: Base Station System
AN: Access Network

BSC: Base Station Controller

CS: Circuit Switched

CS-MGW
MSC server

MSC: Mobile Switching Centre

PSTN

VLR

HSS

HLR

AuC

SMS-GMSC
PS & CS

CN: Core Network

EIR

GGSN

GPRS PS: Packet Switched

Internet
Mobile Station (MS)

- Universal Integrated Circuit Card (UICC)
- Secure Smart Card
- Contains Subscriber Identity Module (SIM/USIM)
- Often: Javacard: Install additional applets (EMV Payment, Ticketing)

Mobile Termination (MT):

- Handles radio transmission, signaling, etc.
- Smartphone: runs on baseband processor (=! app cpu)
Base Station System (BSS)

- Handles all (server-side) radio communication
- Base Transceiver Station (BTS) – 2g, Node B – 3g
  - handles radio communication
- Base Station Controller (BSC) – 2g, Radio Network Controller (RNC) – 3g
  - Controls $\geq 1$ BTS
  - Terminates Link encryption
Base Station System (BSS)

- **Base Station Controller (BSC)** – 2g, **Radio Network Controller (RNC)** – 3g
- **Connect (SS7 signaling) to Core Network / Network Switching Subsystem**
- **Calls via Media Gateway (MGW) to circuit switched Mobile Switching Center (MSC)**
- **Data / SMS via Serving GRPRSS Support Node (GGSN) to packet switched network**
Mobile Switching Center (MSC) routes calls to other MSCs; to the Public Switched Telephone Network

GPRS Support Node (GSN) routes data to the Internet / SMS to Short Message Service Center (SMSC)
Visitor Location Register (VLR)
- In serving network
- Keeps track of currently connected MS

Home Location Register (HLR)
- In subscriber’s home network
- Keeps track of current location of subscribers
Core Network (CN)

- **Home Location Register (HLR)**
  - Provides authentication data / link encryption key to serving network via Authentication Center **AuC**

- **Authentication Center (AuC)**
  - Holds shared secret $K_i$ for each SIM
  - Generates authentication data and link encryption key for each session
Core Network (CN)

- Equipment Identity Register (EIR)
  - Holds globally unique identifiers of stolen, banned, or defective mobile phones
- Unique identifier of MS devices: International Mobile Station Equipment Identity (IMEI)
  - Globally synchronized database

Diagram showing the core network components including PSTN, GMSC, HLR, AuC, HSS, VLR, SGSN, GGSN, and Internet.
Data in SIM Application

SIM / USIM application on UICC contains

- Shared secret $K_i$ (with AuC)
- International Mobile Subscriber Identifier (IMSI)
  - Needed to look up $K_i$, calculate auth data and session key
- Temporary Mobile Subscriber Identifier (TMSI)
  - Stored at VLR together with IMSI
  - Mask IMSI against passive eavesdropping attacks: limited location privacy
Authentication in 2G/3G

Security Capabilities
- TMSI
  - Identity Request
    - IMSI
  - RAND, SQN, AUTHN
    - XRES'
  - Security mode command
    - check MAC

Key = f(K_i, RAND)
XRES = g(K_i, RAND)

AUTHN = h(K_i, SQN, RAND)

3G only

MAC (allowed crypto, security cap)
Two different location privacy attacks

- **Monitorings**: Retrieve identities at a location
- **Tracking**: Retrieve a person’s location

  - Network of Antennas
  - Triangulation

Furthmore:

- **Passive Attack**: Limited Protection from TMSI (does not change often)
- **Active Attack**: Send Identity Request Message. (Prior to authentication)
MitM via 3g to 2g Downgrade

First phase

- Attacker impersonates phone
- Attacker queries currently valid authentication data
- Obtains (RAND, SEQ, AUTHN)

Second phase:

- Attacker impersonates serving network (2g)
- Attacker sends (RAND, SEQ, AUTHN) to phone
- Attacker chooses no or weak encryption (A5/1, A5/2)
- A5/1, A5/2 can be broken in seconds
- Attacker establishes valid connection to network
- Attacker forwards call, sms, data; has plaintext
Conclusion Mobile Network Security

- Security GSM/UMTS/LTE completely broken
- Always use end-to-end encryption for sensitive information
  - TLS Certificate Pinning
  - Signal
- Beware 2-factor Authentication via SMS (mTan, etc)
- SS7 attacks can be launched from anywhere with modest budget
Android Security
Reversing Engineering Android/Java

- Java/Android decompilers
  - Easily available and mostly free
  - Source close to original
- Client side code is not protected by itself

Why?
- Improving applications
- Make it fit for other use cases
- Want to know what is running on your devices
- It is fun
Basic Tools

- Smali/baksmali
  - Assembler/disassembler (language) for Android
  - Easy to modify

- Apktool
  - Disassemble apks to smali
  - Assembling yields valid apk again

- Decompilers (jadx, Bytecode Viewer, ...)
  - Decompile jars and apks to Java
  - Rarely complete decompilation, still useful for understanding logic
Other Techniques

- Hooking, instrumentation, and debugging at run time
- Sniffing network traffic
  - Bypass transport layer protection
  - Harder with Certificate Pinning
- File I/O
- API and system calls
- Log messages, debug output
Why Obfuscate?

- Complicate reverse engineering and exploiting
- Digital Rights Management
- Protect intellectual property (algorithms, data, ...)
  - Especially on client devices
- Certification
Basic Obfuscation Techniques

1. Identifier Remapping
2. Literal Encryption
3. Code Encryption/Packers
4. Noise/Dead Code

- ProGuard does 1., third party software for others
Identifier Remapping

```java
class ExampleClass {
    String greet(String person) {
        return "Hello " + person;
    }
}

class a {
    String b(String c) {
        return "Hello " + c;
    }
}
```

- Used as optimization not obfuscation by ProGuard
- Usage patterns, signature, and code structure preserved
**Literal Encryption**

```java
class a {
    String b(String c) {
        return "Hello " + c;
    }
}
```

```java
class a {
    String b(String c) {
        return d("REDA
        + c;
    }
}
```

- Decryption key stored somewhere
- Literals decrypted at run time
# Packers

- **Compile time:** Code is encrypted and replaced by Unpacker
- **Run time:** Unpacker loads encrypted code, decrypts and runs it
- Off-the-shelf decompilers will not work
- Decryption key stored somewhere
- Code decrypted at run time
Noise/Dead Code

- Instructions not effecting outcome and behaviour
- Unused code paths
- Unused variables
- Arbitrarily complex
- Beware of side effects

- Reverse engineers must distinguish signal from noise first
Debugging vs. Instrumentation

- **Debugging**
  - Debugger attaches to running program
  - Break on instructions
  - Read registers, memory, ...

- **Instrumentation**
  - Instrumentation engine injected into running program
  - Read registers, memory, ...
  - Run arbitrary code at arbitrary location
Instrumentation > Debugging

- Debugging functionality can be simulated with Instrumentation
- Additionally:
  - Change existing code dynamically
  - Dump and follow traces
  - Change method calls, parameters, jumps, ...
Ideas

- Skip password check
- Dump memory at runtime
- Change cryptographic material
- Read and change network traffic
- Circumvent obfuscation
Frida

- Dynamic Binary Instrumentation (DBI) framework
- Windows, Mac, Linux, Android, iOS, QNX
- Scriptable with JavaScript

https://frida.re/
**Literature / Links**

- Meyer (2004): A Man-in-the-Middle Attack on UMTS
- Wehrle (2009): Open Source IMSI Catcher (Masterarbeit)
- Weinmann (2012): Baseband Attacks (WOOT’12)
- Dabrwoski (2014): IMSI-Catch Me If You Can (ACSAC’14)
- Broek (2015): Defeating IMSI Catchers (CCS’15)
- Golde (2012): Weaponizing Femtocells (NDSS’12)
- Jover (2019). The current state of affairs in 5G security and the main remaining security challenges (arXiv)
The vector drawings in slide 32-39 are licensed under GPLv3, the sources are available at https://security.inso.tuwien.ac.at/downloads/ws19/advsecsyseng/gsmstructure/
Thank’s for your attention!

https://security.inso.tuwien.ac.at/