
UNIVERSITY OF FREIBURG

DEPARTMENT OF COMPUTER SCIENCE
CHAIR OF COMMUNICATION SYSTEMS

MASTER THESIS

# IMSI CATCHER DETECTION SYSTEM

*Author:*
Thomas Mayer
tom.f.mayer@gmail.com

*Supervisor:*
Prof. Dr. Schneider
Dennis Wehrle
Konrad Meier

June 11, 2012

# Declaration

I hereby declare that this thesis has been composed by me without any assistance and I have not used any sources or tools other than those cited. Furthermore, I declare that I have acknowledged the work of others by providing detailed references. I also declare that this thesis or parts of it have not been accepted in any other previous application for a degree, project or examination.

Freiburg, June 11, 2012

**Abstract:**

For several years now security flaws in the GSM protocol have been known and exploited. A device called IMSI catcher, first developed in 1996, uses some of these flaws to enable the operator to localise a mobile subscriber and tap into phone calls. Since only authorities were able to obtain these devices, the risk for abuse was deemed minor at first. However, due to the progress in freely available GSM related software and hardware, like OpenBTS and the Universal Software Radio Peripheral, it is now possible for anyone to build an inexpensive version of an IMSI catcher. Although operation is prohibited by law, the possibility of affordable self-construction increases the risk of abuse in the private sector and in relation with industrial espionage. Additionally, operation is near impossible to discover in retrospect.

The goal of this project is to find means and methods of uncovering IMSI catchers that are active in the close perimeter. To that end, the behaviour of such devices and the differences compared to legitimate base stations will be presented and analysed. These findings will then be used to implement the IMSI Catcher Detection System, a tool with a user friendly graphical interface to gather, analyse and visualise information. Evaluations against an IMSI catcher shows the effectiveness of the methods used by uncovering several realistic attacks. The system itself builds upon an open source framework and harvests information about potential IMSI catchers while being invisible itself.

**Zusammenfassung:**

Seit einigen Jahren werden bekannte Sicherheitslücken im GSM Protokoll ausgenutzt um Angriffe durchzuführen. Der IMSI-Catcher, ein 1996 entwickeltes Gerät, benutzt einige dieser Lücken um MobilfunkteilnehmerInnen zu lokalisieren und ihre Anrufe abzuhören. Da solche Instrumente nur für Behörden zugänglich waren wurde das Missbrauchsrisiko als gering eingeschätzt. Weiterentwicklungen im Bereich frei erhältlicher Soft- und Hardware im GSM Bereich, wie etwa OpenBTS oder das Universal Software Radio Peripheral, haben es möglich gemacht einen solchen IMSI-Catcher mit vertretbaren Kosten selbst zu bauen. Obwohl der Gebrauch solcher Geräte gesetzlich verboten ist, erhöht die Möglichkeit des kostengünstigen Eigenbaus eines IMSI-Catchers das Missbrauchsrisiko im Privatbereich oder im Bereich der Industriespionage enorm. Erschwerend kommt die Tatsache hinzu, dass der Einsatz kaum nachvollziehbar ist.

Ziel dieses Projektes ist es Vorgehensweisen zu finden, die den Betrieb eines IMSI-Catchers in der Umgebung aufdecken. Um dies zu erreichen wird das Verhalten eines IMSI-Catchers analysiert und mit dem Verhalten einer legal betriebenen Basisstation verglichen. Mit Hilfe dieser Ergebnisse wird dann das IMSI-Catcher Detection System entwickelt, ein Programm mit einer benutzerfreundlichen Oberfläche, das dazu dient Informationen zu sammeln, auszuwerten und anzuzeigen. Auswertungen von Versuchen zum Auffinden echter IMSI-Catcher, in verschiedenen realen Angriffsszenarien, zeigen die Effektivität der eingesetzten Methoden. Das System selbst baut auf einem open source Framework auf, das es ermöglicht Informationen von IMSI-Catchern zu empfangen und dabei selbst unentdeckt zu bleiben.

**Acknowledgements:**

This thesis would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this project.

First and foremost, I want to thank my supervisor Dennis Wehrle for sacrificing a considerable amount of time reading and annotating my drafts. His constructive comments were of the utmost help in improving the quality of this document. He also gave me valuable hints and new ideas while discussing the methods used in this project. I also want to thank Konrad Meier for helping me out whenever I was stuck, especially with the programming part in the OsmocomBB framework.

My gratitude also goes to the Chair of Communication Systems and Prof. Schneider for providing this interesting topic and all the expensive shiny toys and infrastructure I needed to complete this research.

I also wish to thank my mother, her husband and the rest of my family for constantly supporting me throughout the course of my studies.

Last but not least, I want to express my gratitude to my invaluable friends that I got to know during the last seven years. Thank you for always being there when I needed you and for the great projects that we finished together.

*To my late father*
*who taught me that the best way to learn is getting hands-on experience.*

# Contents

# 1. Introduction

## 1.1. Motivation

Boundless communication for everyone, everywhere, any time. That was the main idea and dream behind the development of the Global System for Mobile Communications (GSM) technology. Considering its reception and growth it can be said that GSM was one of the most successful technologies of the last 30 years [10, 15, 14]. The advent of portable radio equipment and microprocessors in the 1980's made mobile phones technologically possible.

From that point on commercialisation started with more and more providers emerging. With more users, security became an ever more important aspect, since confidential telephone calls were now made over radio instead of fixed landlines. An inherent problem of the air medium is that anybody with suitable equipment can access radio waves, while with landlines physical access is required. In 1996 Rhode & Schwarz released the IMSI catcher [13], a device that takes advantage of security flaws in the GSM protocol which enables it to record phone calls and track users. The name refers to the International Mobile Subscriber Identification (IMSI) number, a unique identification of the user inside the GSM network. It can be obtained by the device by impersonating a base station which is the entry point of the subscriber into the network. To the mobile phone used by the subscriber there is no difference between a real base station and an IMSI catcher. It will always connect to the strongest base station available. By means of a classical man-in-the-middle attack the IMSI catcher operator lures the subscriber to connect to the device and relays the information to a legitimate base station while harvesting the desired information, like calls or IMSI numbers. This process is completely invisible to the user.

Up until now countermeasures to IMSI catchers have not been given much attention to, since the commercial grade devices were only available to authorities and private abuse was thus not a important issue. This risk is intensified by the fact that several other projects like the Open Source IMSI-Catcher [27] demonstrated that such an IMSI catcher can be built at very low cost, using hardware and software that is freely available. It is now possible for anyone to self-construct these devices in an cost-effective manner. With these systems it is considerably easier to eavesdrop on and thus breach the privacy of a neighbour, wife or husband. In the context of industrial espionage, corporate phone calls done over a mobile phone are also easier to target this way.

The detection of illegal private use of IMSI catchers is where this project is aimed at. Different ways will be explored on how to identify an IMSI catcher based on its differ-

ences to a regular base station. In particular, information about the surrounding area and tracking of different parameters over time is used to isolate suspicious base stations in the perimeter. We present a tool that makes it possible to gather and analyse information from all available base stations in an easy manner, using a sophisticated graphical interface: the IMSI Catcher Detection System (ICDS). It also allows switch to an end user mode, where only a very simplified version of the graphical interface is presented and the program evaluates whether it is safe to place a phone call or not. The tool operates in a passive manner, i.e. it only uses information that is freely broadcasted, never connecting to base stations in question. This way the system itself stays invisible to base stations and thus potential IMSI catchers while evaluating them.

## 1.2. Structure

The remainder of this thesis is structured as follows: the second chapter will give an overview of how a GSM network is built up to create a general understanding of the infrastructure in which an IMSI catcher and the detection system are situated. Protocol specifics of the interface on which the two systems operate, the $U_m$ or air interface will be discussed in the second part. The chapter concludes with a description of how an IMSI catcher works and gives an account of what kinds of attacks are possible.

In the third chapter, the software framework and hardware is introduced on which the ICDS is built upon. The different procedures used for information gathering and evaluation are also discussed in this chapter based on possible attacks an IMSI catcher can perform as well as the differences in parameters to a valid base station. Finally a explanation of how to set up and operate the system together with some use cases is outlined.

The fourth chapter contains an evaluation of how the system performs in several categories. First, some general performance statistics and results on the individual methods used are collected. Afterwards, a longer test is conducted over the course of one week to see how well the databases the system uses work in a potentially changing environment. The chapter ends with two simulated attack scenarios.

In the last chapter, a short summary of the results will be given as well as an outlook of how the system can be extended in several ways.

## 1.3. Disclaimer

While conducting the practical part of this thesis precautions have been taken not to interrupt or influence radio transmissions made by regular subscribers. The main part of the experiments is passive information gathering which only harvests information that is freely available and thus does not influence regular communication procedures.

Whenever the IMSI catcher was used, it was configured in a way to not let subscribers connect. Therefore, it is not interfering with regular connection procedures. Operation of the IMSI catcher and the OpenBTS base station were restricted to ARFCN 877 which is officially registered to the university.

## 1.4. On Typesetting

To make the thesis more readable, a few conventions will be kept throughout this document. Important words or components of the ICDS are printed *emphasised* when they first appear. `Typewriter` is used whenever a program or a file name are used in the running text. Code examples can be distinguished by a code listing box that surrounds them.

```
if __name___ == '__main__':
    print 'Hello ICDS'
```

If a complete command line is given it will be put into a new line and the `typewriter` font will be used.

```
sudo do_it -t now
```

Generally a lot of acronyms will be used due to the nature of GSM and telephony dialects, where every possible word has an abbreviation associated with it. The first appearance will always be written out followed by the acronym in parenthesis that will be used from that point henceforth. A complete list of all acronyms for reference can be found in the back of the document.

# 2. GSM

This chapter will give a short overview of some important aspects of GSM networks and protocols. The first section presents a brief historical summary on the evolution of GSM and how it came to be what it is today. In Section 2.2 the system architecture with its components and some essential protocol basics will be explained as far as it is necessary to understand which place in the network an IMSI catcher tries to take over. The $U_m$ interface will be described in detail in Section 2.3 since this is our main source for gathering information from IMSI catchers. Section 2.4 will finally explain how an IMSI catcher works and how it replaces the system components, as well as state from a technical and law perspective why these devices have become a threat to all-day privacy.

## 2.1. A Historical Perspective

The acronym GSM was originally derived from *Group Spéciale Mobile*. This committee was part of the Conférence Européenne des Administrations des Postes et des Télécommunications (CEPT), 1982, with the task of developing a pan-Eurpean digital cellular mobile radio standard in the 900 MHz band. In 1986, the frequency range was officially licensed. The foundation of this task group was a direct answer to the development of independent and incompatible analog radio networks during the 1980's. Examples of such networks were the C-Netz in Germany, the Total Access Communication System (TACS) in the UK and Northern Telecomunication (NMT) in Scandinavia.

In February 1987, the committee submitted the basic parameters of GSM. Not long after, in September, the Memorandum of Understanding (MoU) was signed in Copenhagen by 15 members of 13 countries that were dedicated to deploy GSM in their respective home countries. This agreement was the foundation for allowing international operation of mobile stations using the standard interfaces agreed upon earlier that year. CEPT itself was around since 1959 and its members founded the European Communication Standards Institute (ETSI) in 1988. In the same year the committee submitted the first detailed specification for the new communications standard. The acronym was reinterpreted in 1991, after the committee became a part of the ETSI in 1989, to *Global System for Mobile Communications*. The very same year, the specifications for the Digital Cellular System 1800 (DCS1800) were submitted. These were essentially the same specifications translated to the 1800 MHz band and the basis for the USA's 1900 MHz band. Under the umbrella of the ETSI, many Sub Technical Committees (STCs) began to work on differ-

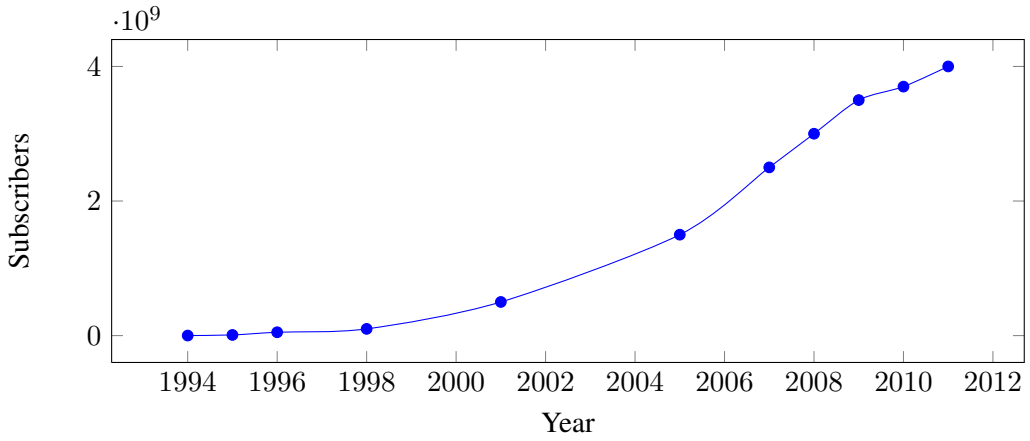



Figure 2.1.: Growth of mobile GSM subscriptions. Compiled from [10, 15, 14]

ent aspects of mobile communication, like network aspects (SMG 03) or security aspects (SMG 10). SMG 05 dealt with future networks and especially with UMTS specifications which eventually became an independent body inside the ETSI.

In 1992, many European countries had operational mobile telephone networks. These networks were a huge success and as early as 1993, they already counted more than one million subscribers [10]. Many networks on different frequency bands (900 MHz , 1800 MHz , 1900 MHz ) were started outside Europe in countries like the US or Australia with Telstra as the first non European provider. The rapid growth of mobile subscribers worldwide until today can be seen in Figure 2.1. Three of the main reasons for this rapid growth are explained by Heine [17] as:

- Liberalisation of the mobile market in Europe which allowed for competition and thus resulting in lower prices and enhanced development.
- Expertise within the Groupe Spéciale Mobile and their collaboration with industry.
- The lack of competitive technologies.

In 1998, the Third Generation Partnership Project (3GPP) was founded by five organisational partners. Their goal was standardisation of mobile communications with focus on developing specifications for a third generation mobile radio system. These partners were the Association of Radio Industries and Businesses (ARIB), the ETSI, the Alliance for Telecommunications Industry Solutions (ATIS), the Telecommunications Technology Association (TTA) and the Telecommunications Technology Committee (TTC). The focus was later expanded in the light of the *International Mobile Communications-2000*-project [9] by the International Telecomunication Union (ITU) to:

- Development and maintenance of GSM and General Packet Radio Service (GPRS), including Enhanced Data Rates for GSM Evolution (EDGE), which are standards for high speed packet oriented data transmission via GSM.
- Development of a third generation mobile communication system on the basis of the old GSM protocol. This standard is called Universal Mobile Telecomunications System (UMTS).
- An IP based multimedia system.

Up to now, the 3GPP has enhanced mobile standards. In 2005 the first High Speed Downlink Packet Access (HSDPA) network went online. HSDPA [8] is a protocol that enables mobile users to download data with speeds of up to 84 MBit / s since release 9. High Speed Uplink Packet Access (HSUPA) [7] is a related protocol in the High Speed Packet Access (HSPA) family that provides similar functionality for uploading data. These and other specification are published on the 3GPP website[1].

## 2.2. The GSM Network

The GSM network is a distributed, star shaped network that is built on top of existing telephony infrastructure to additionally connect mobile users. The telephony network is not only used to connect mobile subscribers to landline phones but also to connect the different components of the mobile network. The main components of a GSM network can be seen in Figure 2.2 as well as the interfaces that are used to connect them. There are different notions of how to distribute these components into functional entities. In the following, the classification by Sauter [23] will be used. It describes the main parts as:

- Basestation Subsystem (BSS): this part is also called radio network and contains all the technology necessary for connecting mobile subscribers to the telephone network and routing their calls. These calls originate from the Mobile Station (MS) that will be explained in Section 2.2.1 and travel over the air interface to the receiver stations for further processing. The air interface or $U_m$ interface will be explained in Section 2.3, whereas the rest of the subsystem will be discussed in Section 2.2.3.
- Network Subsystem (NSS): the core network, as it is sometimes called, consists of several entities that are used to establish and route a connection. This is not only limited to calls within the provider's network but also into other providers' networks or the Public Standard Telephone Network (PSTN). The databases that contain subscriber information and location information for connected users are located here.

[1] 3GPP - Specification Groups, `http://www.3gpp.org/Specification-Groups` [Online; Accessed 04.2012]

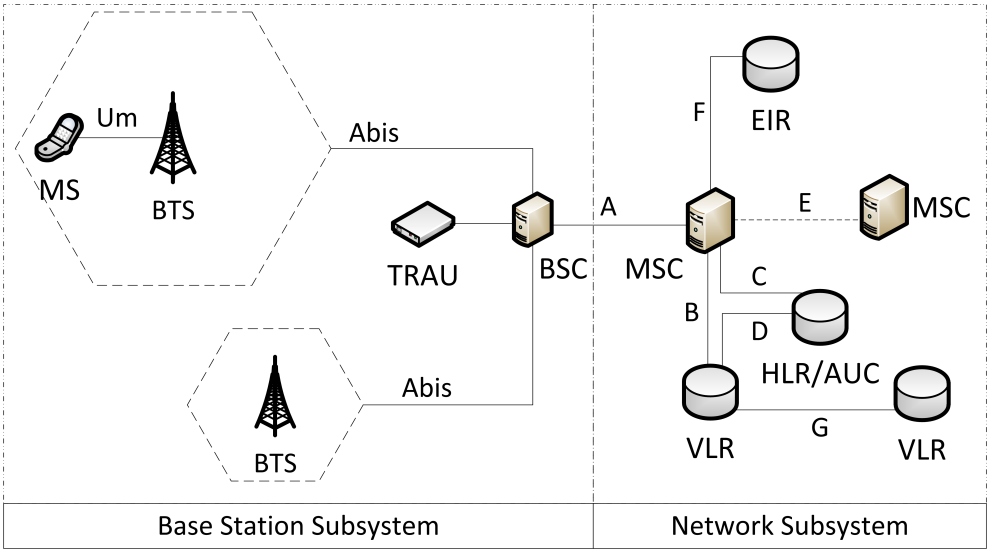



Figure 2.2.: The main components of a GSM network.

- Intelligent Network Subsystem (IN): this part of the network augments the core network with value-added service (VAS) [25]. In order to provide extra functionality, the IN consists of several Service Control Point (SCP) databases. Some of the most widely used services are in fact services of the IN and not core services. Examples are prepaid cards, home areas[1] or telephone number portability.

Other sources define the Operation and Maintainance Subsystem (OMS) [10] or limit the BSS entity to the provider part and define an additional entity for the MS [16, 24]. The system presented in this project works inside the base station subsystem, acting the part of a passive, information gathering MS. Therefore, the following theory section will focus mainly on this part, including the radio interface between the phone and the base station to establish a basic understanding of how the system is able to passively harvest information.

The NSS will only be discussed as far as it is relevant to understanding how an IMSI catcher operates. Since the IN is not involved in any procedure concerning this project, further explanation will be omitted.

### 2.2.1. Mobile Station

With the advent of portable microprocessors in the 1980's mobile phones became technically possible. Advances in technology up to today yielded ever smaller mobile phones

[1]This service defines a geographical area, in which lower rates are calculated for mobile calls.

with ever more functionality. Year by year, this process continued until not the technology itself was the constraining factor for size but the user interface, e.g. button and display sizes. This trend changed with the upcoming of so called smart phones. With weight being the driving factor and not size, resolution and display sizes started to increase again but the devices became ever thinner. What hasn't changed is the basic distinction between Mobile Equipment (ME) and Subscriber Identity Module (SIM), the parts of which a MS consists.

It is hard to deliver a consistent definition for what a ME is. GSM Recommendation 02.07 [11] summarizes the mandatory and optional features of a MS. Some of the most important mandatory features are [17]:

- Dual Tone Multi Frequency (DTMF) signalling capability.
- Short Message Service (SMS) capability.
- The ciphering algorithms A5/1 and A5/2 need to be implemented.
- Display capability for short messages and dialled numbers, as well as available Public Land Mobile Networks (PLMNs).
- A ciphering indicator that shows the user whether encryption is activated on the current connection or not. This feature is disabled in most devices as not to confuse the user.
- Machine fixed International Mobile Equipment Identifier (IMEI). In a strict sense this disqualifies many modern mobile phones since the IMEI is not fixed onto the device itself but is rather part of the software or firmware. Tools like *ZiPhone*[1] for iOS devices[2], especially iPhone, can change this supposedly unchangeable identifier.

A common way to categorise different phones was to group them by the band they support. However, it is more common nowadays that MEs support two bands, three bands or even all four bands. These are called dual-band, tri-band and quad-band devices respectively.

As the name suggests, the SIM card is essentially a data storage that holds user specific data. This separation is interesting for the GSM user since it allows him / her to exchange the ME without having to contact the provider. Thus the same SIM card can be used on different frequency bands which is one of the preconditions for roaming. It card can either be in plug-in format or ID-1 SIM format. The latter one is normally used for telephone cards, credit cards or car installed MEs.

[1] Unlock iPhone 4, Jailbreak iPhone, `http://www.ziphone.org/` [Online; Accessed 04.2012]
[2] Apple iOS5, `http://www.apple.com/ios/` [Online; Accessed 04.2012]

| Parameter | Description |
|---|---|
| Security Related | |
| A3/A8 | Algorithms required for authentication and generation of the session key |
| Ki | Secret key |
| Kc | Session key, generated from a random number and Ki via A8 |
| PIN | Secret numeric password to use the SIM card |
| PUK | Secret numeric password to unlock the SIM card |
| Subscriber Data | |
| IMSI | Subscriber identification |
| MSISDN | Telephone number |
| Network Related | |
| LAI | Identifier of the current Location Area |
| TMSI | Temporary IMSI |
| Home PLMN | Multiple entries to identify the home PLMN |

Table 2.1.: Subset of data stored on a SIM card. Adopted [17]

A subset of parameters stored on the Electrically Erasable Programmable Read-Only Memory (EEPROM) of the card can be seen in Table 2.1. The most important information stored on a SIM card are the IMSI and the Secret Key (Ki).

This key is used to generate the Ciphering Key (Kc), as will be explained in Section 2.2.2. Most of this data, although not the security relevant Ki and Kc, can be read via a USB SIM card reader which can be bought for around $10 on the web. Since Ki never leaves the card, Kc has to be dynamically generated on the card. This can be done since the card itself has a microprocessor that manages the security relevant data. Key functions, like running the GSM key algorithm, verifying a Personal Identification Number (PIN) or reading a file can be accessed through the microprocessor via a communication protocol. A brief description of the protocol and functionalities can be found in Sauter's book [23].

The IMSI as described in GSM 23.003 [1] uniquely identifies a subscriber. It has at most 15 digits and is divided into three parts, Mobile Country Code (MCC), Mobile Network Code (MNC) and Mobile Subscriber Identification Number (MSIN) of which only the last part is the personal identification number of the subscriber.

$$\underbrace{262}_{\text{MCC (Germany)}} \quad \underbrace{01}_{\text{MNC (T-Mobile)}} \quad \underbrace{9876543210}_{MSIN}$$

The first two groups together are called Home Network Identifier (HNI). The three digit

| Country | MCC |
|---|---|
| Germany | 262 |
| Australia | 505 |
| USA | 310 - 316 |
| UK | 234 - 235 |
| Switzerland | 228 |
| Austria | 232 |
| France | 208 |

| Provider | Country | MNC |
|---|---|---|
| T-Mobile | Germany | 01, 06(R) |
| Vodafone | Germany | 02, 04(R), 09(R) |
| E-Plus | Germany | 03, 05(R), 77(T) |
| $O_2$ | Germany | 07, 08(R), 11(R) |
| Orange | France | 00, 01, 02 |
| Swisscom | Switzerland | 01 |
| A1 | Austria | 01, 09 |

Table 2.2.: Mobile Country and Network Codes. (R) denotes that the MCC is reserved but not operational as of yet, whereas (T) denotes a operational test network.

MCC describes the country, the area of domicile of the mobile subscriber. The MNC is an identification number for the home PLMN. It can either have two or three digits depending on the MCC. It is not recommended by the specification and thus not defined to mix two and three digit MNCs for a single MCC. These country codes are assigned by the ITU in ITU E.212 [26]. An excerpt can be found in Table 2.2. The third part, the MSIN is a number consisting of up to ten digits, which is used for authenticating the mobile subscriber against the network. MNC and MSIN together are called National Mobile Subscriber Identity (NMSI).

### 2.2.2. Network Subsystem

The most important task of the Network Subsystem, or Network Switching Subsystem, is to establish connections and route calls between different locations. This is done by the so called Mobile Switching Center (MSC) that can route a call either to another MSC, into the PSTN or into another provider's network. Apart from routing, the NSS also provides the means to administer subscribers inside the network. Facilities to support this task are the Home Location Register (HLR), the Visitor Location Register (VLR) and the Authentication Center (AuC). These will now be described in further detail. A possible arrangement of these components is displayed in Figure 2.2. The Equipment Identity Register (EIR) shown in the picture can be thought of as a database containing lists with information on whether to allow a particular IMSI access to the network or not.

#### Mobile Switching Center

The MSC is the component that does the actual routing of calls and therefore is the core component of the NSS. It basically works like any other Integrated Services Digital Network (ISDN) exchange device with additional functionality to manage mobility. Since the amount of signalling inside a PLMN would be far too much for a single MSC there

is one for every Location Area (LA). Amongst others, its most important tasks are Call Control (CC) and Mobility Management (MM).

CC entrails registration when the subscriber connects to the network as well as routing the calls or text messages from one registered subscriber to another. This routing can include transmission of calls to landlines or to networks of other providers. MSCs that bind the provider's networks to other providers' networks or the PSTN are called Gateway MSCs.

The above part is also true for pure landline switching centres. What sets a mobile switching centre apart from these is called MM. Since the participants can freely move around the network and thus cannot be identified the same way as a fixed landline participant, authentication before using the offered services is important. Another consequence of mobility is that the network has to keep track of where a subscriber is and through which MSC it can be reached. This is done via *Location Updates*, which update the current location in the databases for other MSCs to look up. Also during active calls, if the subscriber leaves the respective service area of the switching centre, the call needs to be transferred to the new switching centre without being interrupted. A procedure called *Handover* achieves just that.

For this central role to work it is necessary to be connected to all the other components of the NSS. This is done via different connections called interfaces. A brief description of what the different interfaces in a GSM network are and what their respective function is can be seen in Appendix A.1.

#### Home Location Register

The HLR is the central database in which all subscriber related data is stored. The entries can be divided into two classes, permanent administrative and temporary data. Part of this administrative data is, which services a subscriber has access to and which are prohibited (e.g. roaming in certain networks). The data itself is indexed with the customer's IMSI to which multiple telephone numbers can be registered. Since these Mobile Subscriber Integrated Services Digital Network Numbers (MSISDNs) are independent from the IMSI a subscriber can change his telephone number and also take the telephone number along should he / she decide to switch to a new provider.

Access to basic services is stored inside the HLR. Examples of such services are the ability to receive and initiate telephone calls, use data services or send text messages. Additional services called Supplementary Services like call forwarding or display of phone numbers during calls can also be set or unset in this database. It is up to the provider if these services are available freely or are bound to a fee. The temporary data enfolds the current VLR and MSC address as well as the Mobile Station Roaming Number (MSRN), which is essentially a temporary location dependent ISDN number.

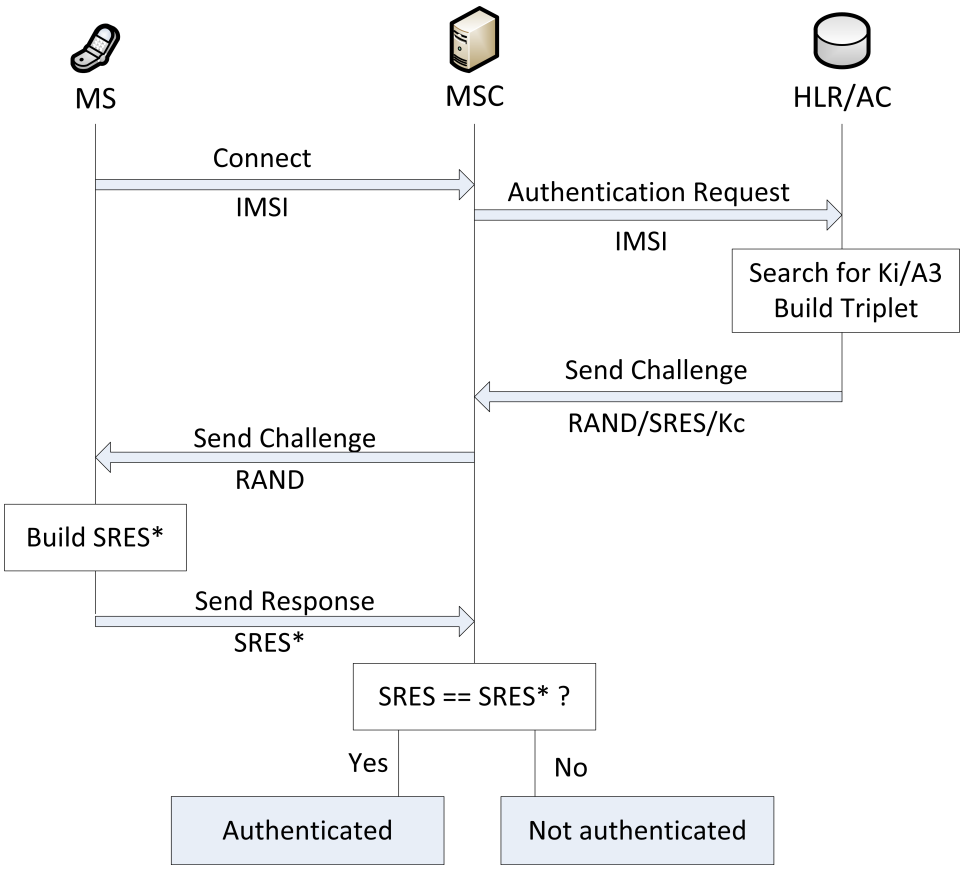



Figure 2.3.: Authentication procedure.

### Visitor Location Register

As can be seen in Figure 2.2 there can be multiple VLRs, one for each area in a network. These registers can be seen as caches for data located in the HLR. They are intended to reduce signalling between the MSC and the HLR. Each time a subscriber enters a new area that is serviced by a new MSC, data for this subscriber is transferred to the respective VLR from the central HLR. Such data includes the IMSI and the MSISDN as well as information on which services are available to that particular subscriber. Additionally, the subscriber is assigned a temporary replacement IMSI called Temporary IMSI (TMSI) and the LA in which the MS was registered last is transmitted. In this way, the regular IMSI is not used and as a result can not be harvested by tapping into the radio channel. While it is possible to operate the VLR as a standalone entity, in most cases it is implemented as a software component of the MSC [23].

### Authentication Center

The AuC is the network component responsible for authenticating mobile subscribers. It is a part of the HLR and the only place apart form the customer's SIM card where the secret key Ki is stored. The authentication is not only done once when the subscriber connects to the network but rather on many occasions, e.g. the start of a call or other significant events to avoid misuse by a third party. This authentication routine is a key based challenge-response procedure[1] outlined in Figure 2.3. The steps of the procedure can be summarized as follows:

1. The user connects to the network or triggers an event that needs authentication at the MSC. There are two possible scenarios from here on.

   In the first case the IMSI is part of the authentication request and the AuC starts with searching for the corresponding Ki and authentication algorithm A3. An authentication triplet is built using Ki which consists of the components:

   - RAND: a 128 bit random number.
   - SRES: a 32 bit number, called signed response, which is generated by A3 with Ki and RAND as inputs.
   - Kc: the ciphering key that is used to cipher the data during transmission. It is also generated with Ki and RAND using the algorithm A8.

   To save signalling bandwidth usually more than one authentication triplet is generated and returned to the MSC by the AuC. It should be noted that, since a separate ciphering key Kc is used, the secret key never leaves the AuC.

   In the second case either a previously generated authentication triplet is used or new authentication triplets are requested.

2. RAND is transmitted to the MS by the MSC where the signed response SRES* is created by the SIM card using A3, Ki and RAND.

3. An authentication response containing SRES* is sent back to the MSC.

4. If SRES and SRES* match, the subscriber is authenticated.

Remarkable properties of this procedure are that by using a ciphering key that is generated by a random number and a secret key, the secret key itself never leaves the AuC. Apart from that, the use of a random number prevents replay attacks on SRES. It should also be noted that this way of authenticating only works for authenticating the subscriber to the network. It is a one way authentication. The subscriber needs to trust the network. This is the basic design flaw that IMSI catchers abuse. In UMTS networks that flaw

[1] A procedure where one party poses a question, a so called challenge and the party to be authenticated has to provide a valid answer.

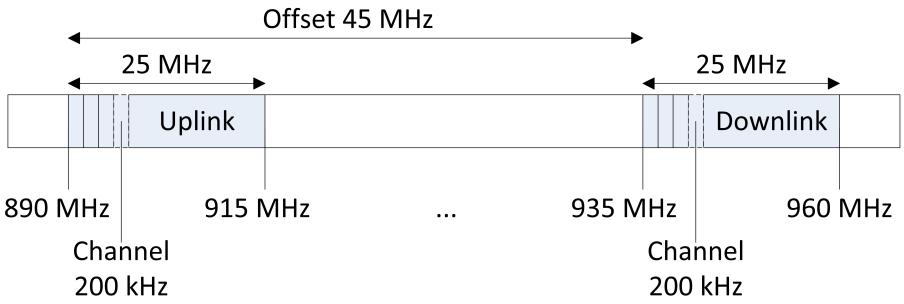



Figure 2.4.: Mapping of functional entities on the 900 MHz band.

was fixed and the authentication procedure was made mutual [23]. However since it will take considerable time until all areas are services by UMTS, phones still have a fallback mechanism to use GSM if no UMTS station is available.

### 2.2.3. Base Station Subsystem

The BSS is the part of the network that provides the hard- and software for physically connecting MSs to the provider's network. Its main components are the Base Station Controller (BSC), the Base Transceiver Station (BTS) and the Transcoding Rate and Adaption Unit (TRAU). Connecting a mobile subscriber works via radio which is why this subsystem is sometimes also called the radio network [23]. Inside the radio network of a certain area, there is one BSC that connects to multiple BTSs and one more TRAUs depending on whether the TRAU is attached to the BSC or to all the BTSs. While the transceiver station acts as receiver for radio signals the controller coordinates the different receivers and relays the incoming signals to the core network. Since signals inside the core network are transmitted at other rates than in the radio network, rates need to be adapted which is done by the TRAU.

Before discussing the individual components of this subsystem it is important to understand how the frequencies of the radio network are used and what architectural impacts this sparse resource has on the network and the components itself.

#### Frequencies and the Cellular Principle

A frequency band as shown in Figure 2.4 is distributed into different functional entities. The band is divided into a range for the uplink, the part that is used by the MS to upload data into the network and the downlink that is utilised by the network to send data back. In the 900 MHz band each of these has a width of 25 MHz . These bands themselves are furthermore divided into channels, each spanning 200 kHz, which accounts for 125 channels on 25 MHz .

| Band | ARFCN | Uplink (MHz) | Downlink (MHz) | Offset (MHz) |
|---|---|---|---|---|
| GSM 900 (Primary) | 0-124 | 890-915 | 935-960 | 45 |
| GSM 900 (Extended) | 0-124 975-1023 | 880-915 | 925-960 | 45 |
| GSM 1800 | 512-885 | 1710-1785 | 1805-1880 | 95 |
| GSM 1900 (North America) | 512-810 | 1850-1910 | 1930-1990 | 80 |
| GSM 850 (North America) | 128-251 | 824-849 | 869-894 | 45 |

Table 2.3.: Frequencies in the different bands [23].

Each of which is identified by its Absolute Radio Frequency Number (ARFCN). This is a simple numbering scheme, given to those 200 kHz channels. The frequencies and ARFCNs are connected as follows:

$$F_{\text{Uplink}} = \text{Band}_{\text{Start}} + 0.2\,\text{MHz} \cdot (\text{ARFCN} - \text{ARFCN}_{\text{Start}}) \tag{2.1}$$

$$F_{\text{Downlink}} = F_{\text{Uplink}} + \text{Band}_{\text{Offset}} \tag{2.2}$$

In case of the 900 MHz Band this would be:

$$F_{\text{Uplink}} = 890 + 0.2 \cdot (\text{ARFCN} - 0) \tag{2.3}$$

$$= 890 + 0.2 \cdot \text{ARFCN} \tag{2.4}$$

$$F_{\text{Downlink}} = F_{\text{Uplink}} + 45 \tag{2.5}$$

For other bands the numbers differ but the functionality is the same. They can be seen in Table 2.3 along with their respective ARFCN numbers

An additional method called time multiplexing, which will be explained in further detail in Section 2.3, makes is possible to map $125 \cdot 8 = 1000$ channels onto that band that could be used for voice transmission. Some of these channels need to be used for signalling. Even though the number by itself seems high, it would never suffice to service a large urban area. This is one of the reasons why another frequency band in the 1800 MHz range has been opened with 75 MHz up- and downlink supporting 375 channels. That by itself would also never suffice to service the huge number of subscribers, therefore, the GSM network like any other modern mobile radio network is based on a cellular architecture which makes it possible to reuse frequencies. The range of one receiver station is drastically reduced to service only a small area. This is called the cell of the BTS which in theory can be approximated by a hexagon, each of which has its own Cell Identities (CIDs). Each of these cells is assigned a different frequency to avoid interference. However, after a certain distance, the *frequency reuse distance* $D$, is covered

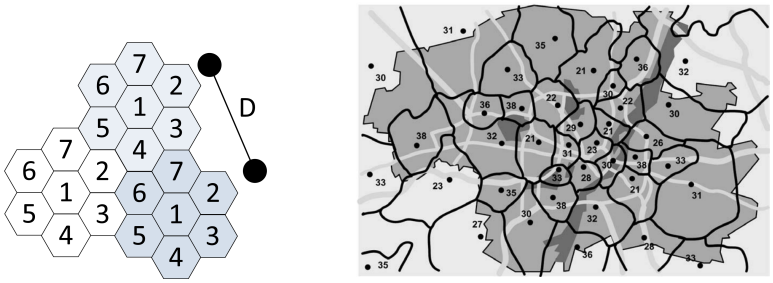



Figure 2.5.: Theoretical arrangement of radio cells compared to a realistic alignment. Cells with the same number share the same frequency [10].

the exact same frequency can be used again by another BTS. $D$ is chosen large enough, so that interference does not have an impact on overall call quality. Figure 2.5 shows such an arrangement. Also, a comparison with realistic cells can be seen which differ in their appearance from the optimized hexagon model. The borders are blurred because of interference, reflection- and shadowing effects and cells in the more urban areas are smaller than cells on the countryside, where the density of subscribers is less and thus can be handled by fewer BTSs. The band has been divided into seven frequencies which are only reused (cells with the same number) after distance $D$ is covered. For an arbitrary division of the frequency band into $k$ partitions and a cell radius of $R$ geometric derivations from the hexagon model yield for the frequency reuse distance $D$ [10]:

$$D = R \cdot \sqrt{3k} \tag{2.6}$$

This procedure raises the number of effectively usable frequencies by a large factor. However certain disadvantages come with this procedure as well [17]. Increasing the amount of receivers automatically increases the cost of infrastructure for the provider. Due to the nature of the mobility of subscribers this increases the amount of Handovers needed since it is more likely that a subscriber leaves a small cell during an active call. These inflict increased signalling load on the network itself.

### Base Transceiver Station

They are also called base stations and are the entry points to the network for subscribers. Theoretically, a BTS can serve a cell of 35 km radius, however, this is decreased by interference, reflection- and shadowing effects. This is the theoretical limit for a cell on the 900 MHz band. A cell on the 1800 MHz band has a lower coverage since the signal falloff is greater due to the shorter wavelength. The limiting factor here are the number of subscribers itself. A single station can only serve a limited number of users which yields a radius as low as 100 m for a single BTS in urban housing areas [23] with high

population density. On the countryside where population is less dense, the constraining factor can be the transmission power of the ME. Therefore, cells with a radius of above 15 km are seldom seen.

BTSs and their corresponding cells can have different configurations depending on load or morph structure of the surroundings. In a *standard configuration* every base station has its own CID, it is a one to one mapping of cells to BTS. This is a cost effective way of providing service to a rural or sparse settled area, since only one BTS is used to cover a large area. For urban densely settled areas, the *sectorised configuration* has become the de facto standard. The main idea is to not have a $360^\circ$ coverage for a base station handling a cell but rather split the cell into multiple sectors, each with its own BTS covering $120^\circ$ for example. This way the amount of subscribers in the cell will be divided over three BTSs instead of one.

### Base Station Controller

The BSC is the central unit in the BSS. It can be compared to a digital exchange in a standard telephone network with additional mobile extensions. The design idea was to remove all radio related load from the MSC into the radio subsystem. Therefore a BSC manages the multitude of BTSs in the BSS.

First and foremost, it is a switching centre. This means it has to switch incoming traffic channels from the MSC over the A-interface to channels on the outgoing $A_{bis}$-interface, which leads over the BTS and thus the air interface to different MSs. As a result, the initialisation and maintenance of signalling and voice channels are its main tasks. What channels are and how they are established is explained in Section 2.3.2. For the sake of functional explanation of the BSC it will suffice to regard channels as a communication line for a particular purpose, like receiving or sending voice data or for sending broadcast information. Due to the nature of a mobile network certain other tasks have to be performed here as well, such as Handovers and power management [23].

A *signalling channel* is needed when a subscriber wants to start a call or send a text message. The MS sends a channel request message to the BSC which needs to check if any Standalone Dedicated Control Channels (SDCCHs) are free. If there are free channels, one of those channels is activated via the BTS and an Immediate Assignment Message (IA) is sent via the Access Grand Channel (AGCH) containing the number of the assigned channel. From this point on the MS can sent data on the assigned channel that reach the MSC. For incoming calls a prior step has to be taken. The MSC sends a message to the BSC that contains the IMSI, TMSI and LA of the subscriber that is being called or texted. This message is forwarded to and broadcasted by all cells in that LA on the Paging Channel (PCH). As soon as this message arrives at the respective MS it requests a channel with the procedure outlined above.

After a signalling channel is found that way, a *voice channel* can be initialised. The MSC sends an assignment request message to the BSC after the start of the call has been

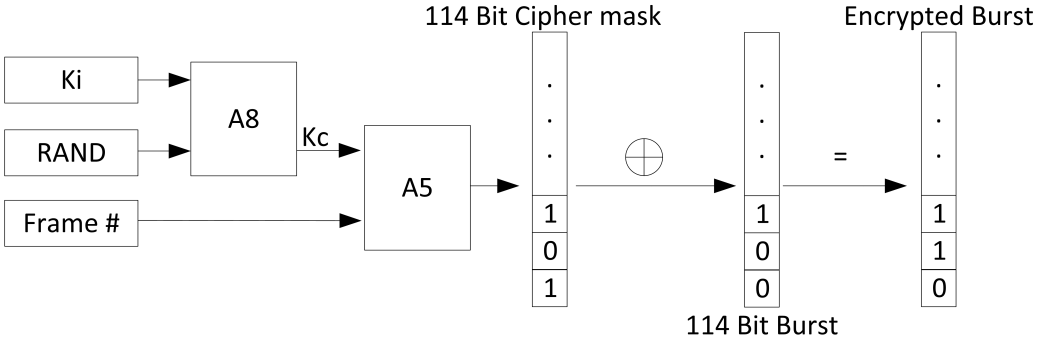



Figure 2.6.: Ciphering procedure for one frame of voice data. Adopted from [23].

determined on the previously assigned SDCCH between the MSC and the MS. A free Traffic Channel (TCH) is assigned and the MS can tune in to this channel and send an acknowledgement to the BSC, which in turn sends an acknowledgement that the assignment has been completed to the MS and the MSC.

Since the voice data is sensitive it is encrypted before it is sent to the NSS. Voice data is a continuous stream originating at the mobile phone and accordingly has to be encrypted using a stream cipher. The stream cipher key $K_c$ is generated by the authentication centre. It is generated by the A8 algorithm on the SIM card with a random number (RAND) and the secret key Ki as input. Since the transmission of voice data is split into frames it suffices to encode the data on a per frame basis. Kc and the current frame number are the inputs for the algorithm A5 which generates a 114 bit ciphering sequence that can be XORed with the frame. This sequence changes every frame since it uses the current frame number as input. The complete procedure is outlined in Figure 2.6. Some strong ciphering algorithms are not permitted in certain countries so there is a variety of algorithms called A5/0, A5/1 and A5/2 from which one needs to be chosen upon connecting to the network. However, the encryption is only optional and not mandatory. The use of A5/0 indicates that no encryption is used. If the network does not offer such encryption, the ME sends its data unencrypted without giving notice to the user in most cases. A ciphering indicator is part of most mobile phones but on normally it is disabled by the operator as to not confuse the customers. The other weakness is the locality of encryption. The procedure only affects the transmission from the ME to the BTS, everything after that is unencrypted voice data. This is especially a problem if providers use point-to-point radio systems to connect their base stations to the MSC.

A *Handover* is necessary when a subscriber leaves the area of a cell and needs to be assigned to another one while conducting a call. First of all a TCH in the target cell has to be activated since the call is still in progress. Once this is done, the new cell address and frequency is sent to the MS over the Fast Access Control Channel (FACCH)

along with a command that triggers the Handover. After synchronising with the new cell, an acknowledgement is sent by the base station to the controller to switch the voice connection to the new cell. What remains is freeing the old TCH for further use by other subscribers.

## 2.3. The $U_m$ Interface

As with all radio based networks, the efficiency of the wireless interface, the interface between the MS and the BTS, is of utmost importance to the overall performance of the network. The main reason is that resources on the air interface are scarce. Maximising efficiency in this case can be seen as maximizing the quotient of transmission rate over bandwidth used [17].

The first section will explain how transmission in a GSM network is handled on the physical level and what techniques are used to maximize throughput. Afterwards, the notion of logical channels, virtual channels that are mapped on top of the actual transmission, will be discussed. It will be carved out which channels are of importance for this project. The last section outlines the network layers of the GSM stack, to give a basis for understanding where the framework employed in the practical part is situated in that hierarchy.

### 2.3.1. Radio Transmission

Without additional techniques, the BTS would only be able to serve a single caller at a time. Therefore, even in older radio networks, like the C-Netz in Germany, Frequency Division Multiple Access (FDMA) is used. With FDMA, a specific frequency of the broad frequency band of the BTS is allocated to a specific subscriber for a call, leaving other frequencies open to be used by other subscribers connected to the same base station. Essentially this means that every BTS can serve multiple frequencies at the same time. This comes at the cost of additional hardware, since all the frequencies need their own transceivers and need to be amplified accordingly to guarantee transmission quality. Additional hardware for each channel is also required to enable duplex transmission, meaning that sending and receiving can be done at the same time.

That number of available frequencies would not suffice to meet the demand, more communication channels were needed. To that end, another technique has been introduced, called Time Division Multiple Access (TDMA). In GSM networks, each of these subbands yielded by the FDMA procedure has a width of 200 kHz. Onto this smaller carrier frequency, TDMA frames are transmitted that contain eight time slots. These frames have a transmission length of 4.615 ms. Each of these timeslots can host the data of a different subscriber, although the first two are usually used for signalling procedures. An illustration of how these multiplexing methods work together can be seen in Figure 2.7.

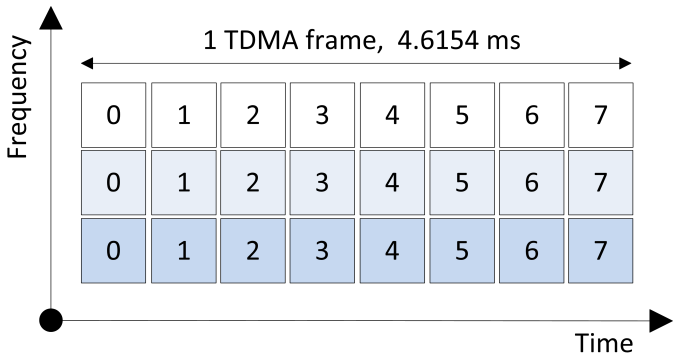



Figure 2.7.: The combination of FDMA and TDMA.

### Frame Numbering

Another important aspect is the frame hierarchy and the resulting frame numbering, since it is used for ciphering as well as channel mapping and synchronisation. The frame number is one of the inputs required to generate the ciphering key and is broadcasted frequently on the Signalling Channel (SCH) to keep mobile subscribers in sync.

An overview of the numbering hierarchy is illustrated in Figure 2.8. The timeslots on the lowest level of the hierarchy have a length of $4.615\,\text{ms} \div 8 = 577\ \mu\text{s}$ and are also known as *Bursts*, numbered from 0 to 7. Depending on what the Burst is used for, the internal structure can differ but the duration is always the same. Every new TDMA frame, the sequence number is increased by one. Since this number cannot be increased endlessly it is repeated every 3 h 28 m 53 s and 760 ms. This is the largest chunk in the frame hierarchy and it is called Hyperframe. Superframes and Multiframes are layers between the Hyperframe and the TDMA frame which can occur in different configurations. The 51-Multiframe consists of 51 TDMA frames and carries only signalling data whereas the 26-Multiframe contains 26 TDMA frames and carries traffic and control channels. Superframes can be seen as packages to wrap these different Multiframes in one packages of consistent lengths.

When a MS and BTS start to communicate the frame number has to be obtained by the MS through the SCH before it can ask for a channel. This is important since the frame number is a vital information, indicating the chronological order of control channels. If the MS asks for a channel assignment in frame $n$ and a channel is assigned to the MS, the assigned channels refers back to the frame $n$ and thus the MS can find its channel amongst the others.

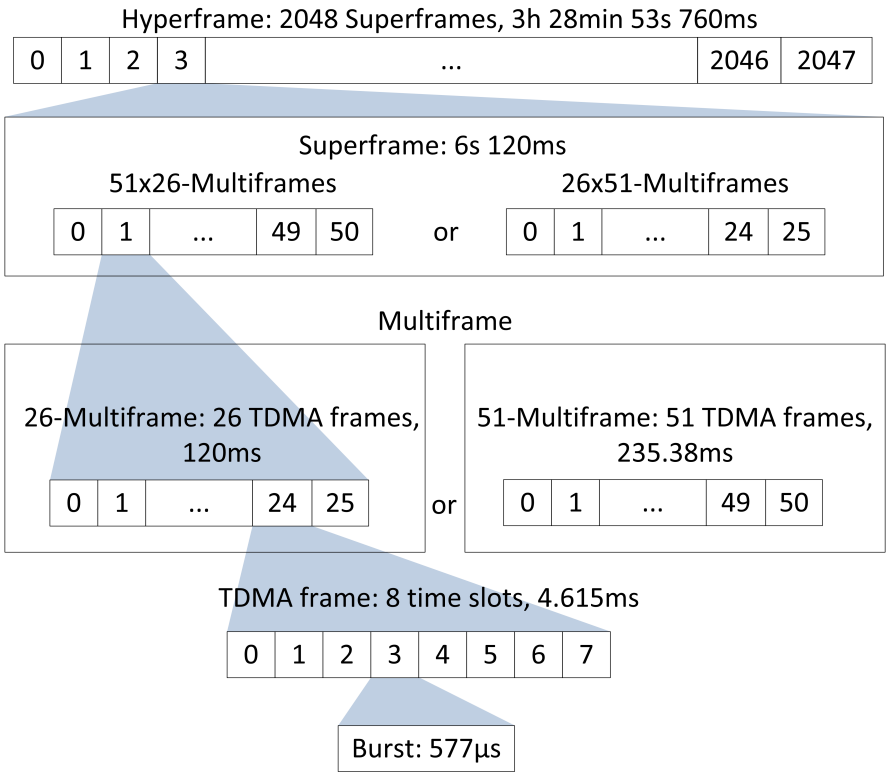



Figure 2.8.: Hierarchical composition of the different frames.

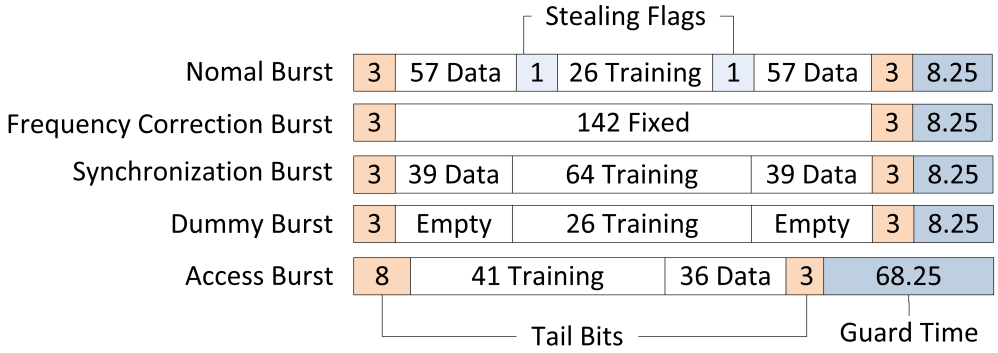



Figure 2.9.: Structural Comparison of different Burst types. After [10].

### Burst Types

As suggested by the paragraph above there are different kinds of Bursts which are shown in Figure 2.9 [10].

In addition to *data bits* and known fixed bit sequences every frame has *tail bits*, which mark the beginning and the end of a frame. The fixed bit sequence is called *training sequence* and appears in conjunction with the data bit sequences. During a radio transmission procedure the signal can be distorted by shadowing, reflection or other factors which would result in a loss of data. But since the training sequence is known, it is possible to reconstruct the original signal by comparing the incoming training sequence with the expected one and thus conserving the data bits.

All Bursts also contain *guard times* which separate them from the next Burst. This is necessary because subscribers can move around and thus slight variations in timing may occur. These variations could result in the collision of data from several different sources, rendering it unusable. For subscribers that move at considerable speeds, e.g. in a car, this is not sufficient and an extra mechanism called *Timing Advance* is used. Basically, the farther a subscriber is away from a base station the earlier a burst has to be sent, to compensate for the distance. The value for the Timing Advance is determined by the BSC after receiving a channel request message from the mobile station and afterwards constantly updated by the respective BTS. The different Burst types are:

- Normal Burst: The basic information transmitting Burst. All information on traffic and control channels is transmitted by this Burst except for the Random Access Channel (RACH). Furthermore, this Burst contains Stealing Flags (SFs). If these are set, the Burst contains important signalling data that has to travel fast over the FACCH, however, no normal data can be transmitted in this case.

- Frequency Correction Burst: This Burst is sent frequently and is used by MSs to fine tune to the frequency of the BTS. It may also be used by the MS to do time synchronisation for TDMA frames. The periodic broadcasting of this Burst forms the Frequency Correction Channel (FCCH) and shares a frequency with the Broadcast Channel (BCCH) as will be shown in the next section.

- Synchronisation Burst: This Burst contains time synchronisation information from the BTS for the MS as well as the running TDMA frame number. Periodic broadcasting of this Burst forms the SCH.

- Dummy Burst: When no other Bursts are sent on the frequency carrying the BCCH, this one is transmitted to fill the gap. This way the MS can keep up doing quality measurements even if no data needs to be transmitted.

- Access Burst: The Burst that is used to transmit data on the RACH. Since everyone

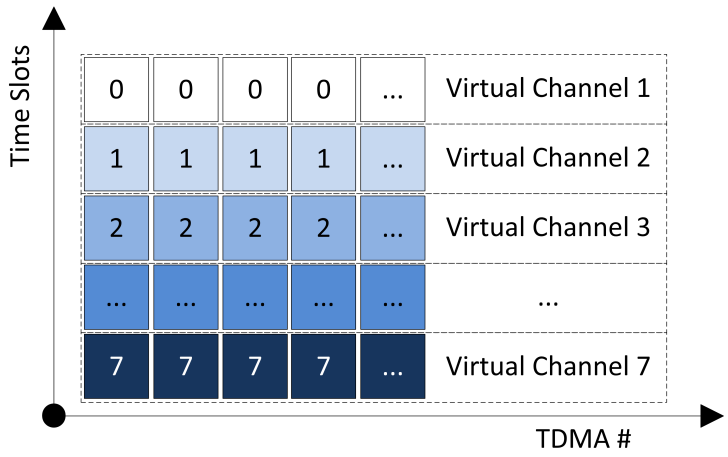



Figure 2.10.: Mapping of virtual channels on time slots.

can sent on the RACH without being given a timeslot via Slotted Aloha[1] procedure, the guard times of this Burst are high as to reduce the probability of data collisions.

The information in this section described the physical properties of the Air Interface. also called Layer 1 when referring to the standard ISO / OSI model. A short description of the other layers will be presented in Section 2.3.3.

### 2.3.2. Logical Channels

A logical channel is a virtual construct on top of the physical construct of frames to group similar information together. Since not all information has to be sent all the time, these different information channels, e.g. broadcast information about the respective base station, can be multiplexed and sent together.

Mapping of these channels on the physical interface works in two dimensions. The first dimension is the frequency and the second is the time slot. Figure 2.10 shows this mapping of channels onto time slots over the course of multiple TDMA frames for one fixed frequency. This way each timeslot over the course of multiple frames can be regarded as a virtual channel. These resulting virtual channels can now be used by a multitude of logical channels to transmit information.

There are two main categories of logical channels, distinguished by their usage [23], dedicated channels and common channels. Dedicated channels transport data meant for a single subscriber whereas common channels contain information interesting to all subscribers.

[1] Slotted Aloha is a medium access procedure in which each participant can send data in predefined timeslots. If collisions occur the data is discarded and each participant has to wait a random time interval before sending again.

### Dedicated Channels

As mentioned above, these channels wrap the communication of a single user with the network. These are point to point channels.

- TCH: A data channels that is used to transmit voice data or data service packages.
- FACCH: A channel for transmission of urgent signalling data, e.g. Handover signalling. This data doesn't have to be send often, so it shares a timeslot with the TCH and uses the stealing flags to insert its own data.
- Slow Access Control Channel (SACCH): The uplink of this channel is used by the MS to transmit quality measurements of the cell and neighbouring cells to the base station, so the network can do Handover decisions accordingly. The downlink is used for Timing Advance data and power management data for the MS.
- SDCCH: On this channel signalling information is sent to a subscriber as long as no TCH has been assigned during the initialisation of a call. Text messages and Location Updates are also transmitted on this channel.

### Common Channels

The common channels contain data interesting to all subscribers, thus having a broadcast nature. These channels are the main source of information gathered by the ICDS. They are point to multi-point channels.

- SCH: When the MS is looking for a cell to connect, this synchronisation channel is used.
- FCCH: It is used by MSs to fine tune to the frequency of a certain base station and helps to find the start of a 51-Multiframe.
- BCCH: This channel is used to transmit information about the network and the base station itself through different *System Information Messages*. These contain the network name and cell identification as well as neighbourhood information on cells in the area and much more. This channel will be the main source of information for this project, since it allows harvesting information without actively participating in the network and will thus be discussed in further detail in Chapter 3.2.1.
- PCH: If a subscriber is not assigned a dedicated channel yet, i.e. he / she is not active, they are notified on this channel if there is an incoming call or text. The subscribers are identified by their TMSI which has been previously assigned upon entering the network. This channel will be used as an additional source of information for the ICDS.

- RACH: A subscriber that has been notified over the PCH can contact the network and request a SDCCH. Since this is a channel used by all connected and idle MSs, access has to be regulated. As the name implies, access is random thus it can happen that two or more MS try to send at the same time. Slotted Aloha is used to handle access.
- AGCH: This is the channel used to respond to a MS if a request has been made on the RACH. The acknowledgement message also contains information on which SDCCH to use.

#### Combinations

These channels cannot arbitrarily be mapped onto Multiframes. There is a complex multiplexing scheme defined in GSM 05.02 [3] that explains which channel combinations can occur inside a Multiframe. A table containing the possible combinations can be found in Appendix A.2. The mapping of these specific Multiframe-configurations onto timeslots is not arbitrary either. Normally TS-0 and TS-1, the first two time slots, are used to handle channels with signalling information. The BCCH for example, which we will use to harvest information uses TS-0 on the carrier frequency.

### 2.3.3. Layers

Design-wise the layers of the $U_m$ interface resemble the layers of the ISO / OSI reference model. This section will give a short overview over the first three layers with respect to the air interface [17], since these are the ones that the employed framework works on.

**Physical Layer (Layer 1):** This layer provides the facilities for the actual transmission of data. In case of the $U_m$ interface, this is the actual radio equipment. On this layer no differentiation between data types like user or signalling data is done. The data that it receives from Layer 2 is either single bit data or an arrays of bits. Gaussian Minimum Shift Keying (GMSK) modulation is used to encode the data a Burst contains into radio signals.

**Data Link (Layer 2):** On Layer 2 packaging is done. The notion of data frames is introduced to have chunks of information on which error checking and potential retransmission of corrupted data can be performed. The Layer 2 protocol High Level Data Link Control (HDLC) is used as a basis for Signaling System 7 (SS-7) as well as for Link Access Procedure, D Channel (LAPD), which are the basic protocols a classical telephone network operates upon. HDLC and its derivatives use start / stop markers and checksums to form data frames. The Layer 2 format changes through the course of the network while the data packages of Layer 3 may stay the same. When a transmission from a MS to the

BTS is done, LAPD Mobile ($LAPD_m$) is used, which is essentially the same as the Layer 2 ISDN protocol with a few simplifications. From the BTS to the BSC, $LAPD_m$ converts to LAPD and afterwards is exchanged to Message Transfer Part 2/SS7 (MTP 2/SS7). For the air interface $LAPD_m$, along with channel coding and Burst formatting form Layer 2. More information about these Layer 2 protocols can be found in the respective Technical Specifications of the 3GPP [2, 5].

**Network (Layer 3):** Layer 3 headers have to provide all the information necessary for the packet to be routed towards its recipient. As with Layer 2 information, it may be the case that this header needs to be partially rewritten during the transmission of a package. Between the MS, BTS, BSC and MSC the Radio Resource (RR) protocol and the information needed to route a call into the SS-7 subsystem are part of Layer 3. This protocol handles configuration and allocation of radio channels as well as managing the dedicated channels to the subscribers.

## 2.4. IMSI Catcher

An IMSI catcher is a device that is used to capture the IMSI and IMEI numbers of mobile subscribers. The knowledge of the IMSI and IMEI numbers can be exploited to either tap into the participant's calls or pinpoint the location of the subscriber [13]. Another less known functionality is that if catchers do not relay intercepted calls they can be used to suppress mobile communication in a certain area, e.g. during a police operation [30].

This topic came up in conjunction with crime fighting and prevention with the advent of mobile telephones. A mobile phone cannot be tapped in the same way as a landline phone, since the subscriber can change places and also phones thus there is no designated line associated with him / her. This has proven to be a challenge to the authorities.

In 1996, Rohde & Schwarz a company based in Munich, Germany has developed a device called *GA 090* which was the first IMSI catcher. Its was capable of yielding a list with all the IMSI numbers in the perimeter as well as pinpointing the location of a subscriber given the IMSI. Short thereafter, the *GA 900* was presented, which had the additional capability of tapping into calls that originated from a particular IMSI. These commercial versions of catchers, produced by Rohde & Schwarz, were priced between 200.000 € and 300.000 € in 2001 [13]. Regulations prohibit the use of IMSI catchers for individuals, because the frequency bands the GSM network uses are registered to providers. In addition to these commercial products, different projects [27, 21] have shown that such devices can be built at a very low budget. This only intensifies the risk that is imposed by the abusive usage of such a catcher. Examples of malicious usage by individuals would be curious neighbours eavesdropping or a jealous husband tapping into phone calls of his wife. On a more large scale, these devices are of great value for industrial espionage or private investigators that would not mind breaking the law

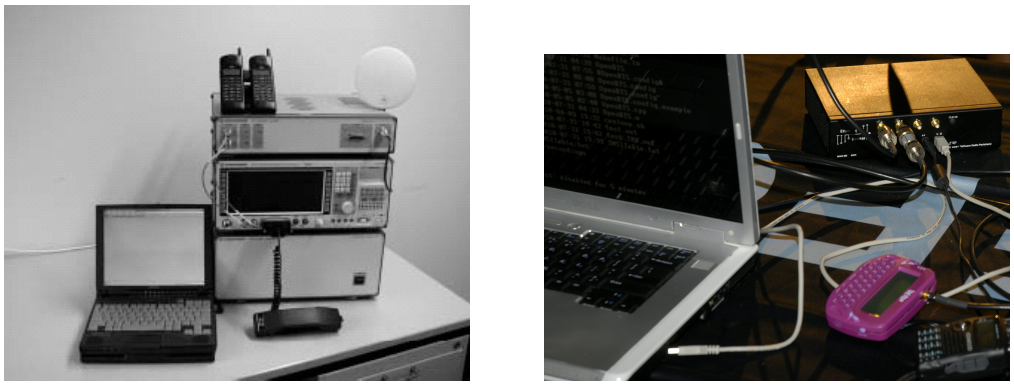


Figure 2.11.: A commercial catcher by Rhode & Schwarz [13] and a self built catcher introduced at Defcon 2010 [21].

to gather information. To uncover such abuse by individuals is the aim of this project. Figure 2.11 shows a commercial model side by side with a self built catcher.

Section 2.4.1 will show how an IMSI catcher works and how subscribers can be caught. In addition the potency of these attacks will be evaluated and what risks these impose from a technical perspective. The next section will explain under which circumstances a catcher can be used in Germany from a legal perspective and show that this handling poses the risk of privacy breach to citizens.

### 2.4.1. Mode of Operation

An IMSI catcher masks itself as a base station and lures subscribers in its perimeter to connect to it without their knowledge. In the attack shown in Figure 2.12 [12], the IMSI catcher is broadcasting a new Location Area Identifier (LAI) with the same CID as an formerly existing base station to the MS, at very high power. This lures the MS to connect to the alleged base station due to stronger reception and announce itself since the Location Area Code (LAC) has changed.

Once a subscriber connects to the device, a command is sent to the MS which asks for the SIM's IMSI. This command is normally only used in case of an error [13] but can be abused this way.

An IMSI catcher can only impersonate a base station because authentication in a GSM network is one-sided as discussed earlier in Section 2.2.2. The subscriber has no way of checking the authenticity of a base station but rather has to trust the broadcasted identifier which can be easily forged by a catcher. At this stage, the subscriber can already be localised as being in a certain distance of the catcher.

In case the IMSI catcher was operated by authorities, they can now query the provider for personal information about the subscriber, however, criminals may use fake creden-

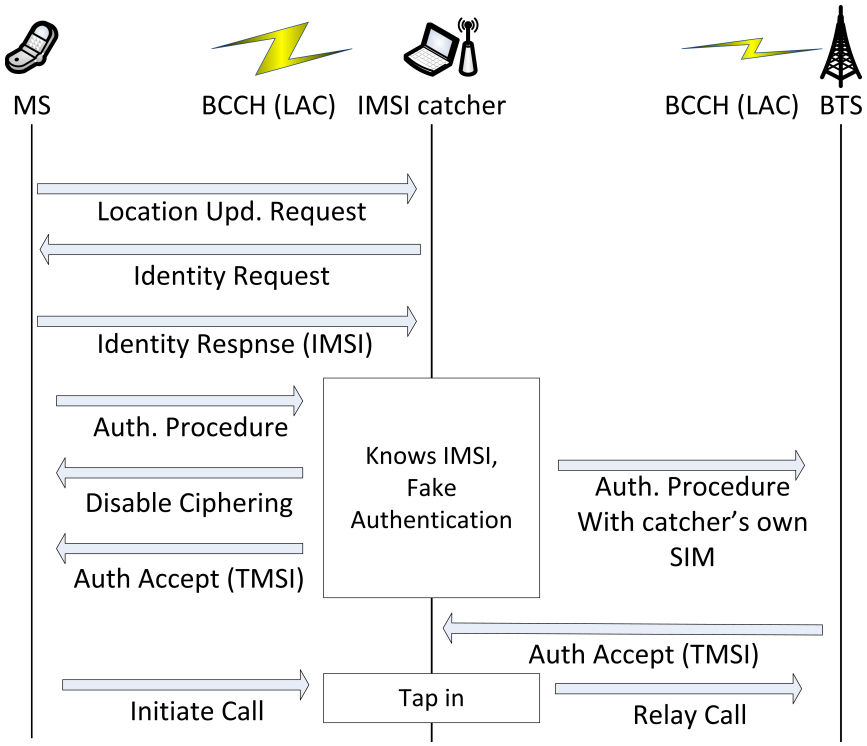



Figure 2.12.: IMSI catching procedure. Adopted and simplified from [12].

tials when obtaining a SIM card. Since it is only possible to catch all the IMSIs in an area, the person to be observed has to be followed and the catcher has to be used multiple times. Each time it yields a set of numbers in the area. The IMSI, that is part of all the sets is the IMSI of the person under observation. More catchers can now be used to triangulate the position. The next step is also possible because of a design decision made in the GSM protocol. Encryption itself or certain kinds of strong encryption are not allowed in all countries. Therefore, it is possible for the base station to request the encryption algorithm A5/0, which means that no encryption will be used for the calls at all. Only a few mobile phones display that encryption has been disabled by the BTS.

At this point the setup for a man-in-the-middle attack [12] on calls is completed. The catcher is connected to the mobile network with its own SIM. If the subscriber now initiates a call, the call can be routed by the catcher into the network and since encryption is turned off, it can also be listened to or recorded. The subscriber doesn't notice this privacy breach, except in the rare cases where the phone displays that encryption has been turned off. The IMEI is also harvested in a similar fashion, if the observed person tries to switch SIM cards on a regular basis [13].

### Attacks

When operating a catcher, the first and most important step is to actually trick the MS into connecting to the catcher. A lot of phones save the frequency they were tuned to last and upon connecting to the mobile network, this is the first frequency they try. Therefore, a MS has to be set to *normal cell selection* mode, which means it starts scanning for the best base station available. Three ways of luring a subscriber to the forged cell were presented by Wehrle for the 'Open Source IMSI-Catcher' project [27]. These methods differ on whether the MS already is in normal cell selection mode or not.

**MS is in normal cell selection mode:** The IMSI catcher has to fake a cell configuration consistent with the provider, the target MS is looking for, broadcasting at any frequency. The MS will choose the base station with the strongest reception levels, so the catcher has to make sure that no other available station has a better reception than itself. Some IMSI catchers even broadcast at a higher power than it would be allowed by law for legitimate BTS [30].

**MS is already connected to a network:** If this is the case then the connection to the current cell needs to be broken or the MS has to be stimulated to switch the cell to the catcher's. A MS that is in passive mode, meaning no active calls are conducted, will do quality measurements on the neighbouring cells of the cell it is connected to. It will not scan for *new* base stations. Therefore, the IMSI catcher has to replace an existing base station that is already part of the neighbourhood of the current cell, so the MS will do power measurements on its frequency. Figure 2.13 illustrates the procedure. In the beginning

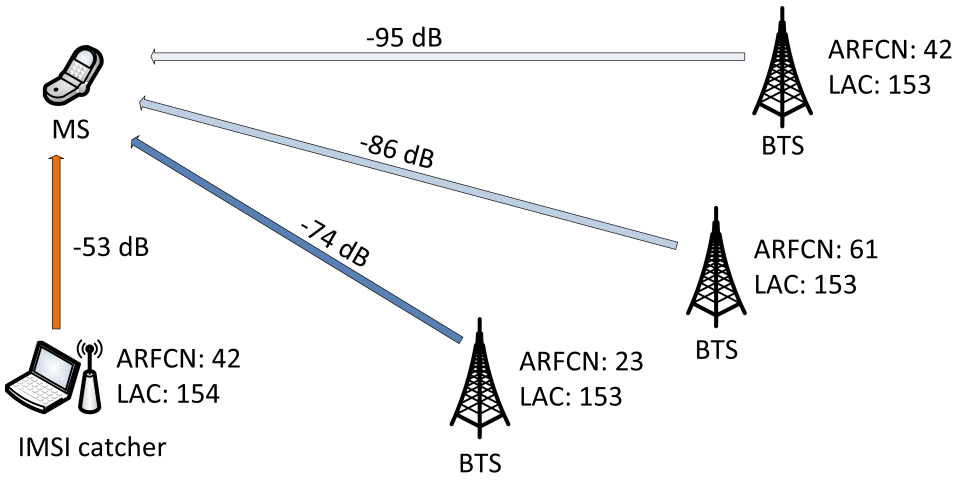



Figure 2.13.: Takeover attack of an IMSI catcher on a base station.

the MS is connected to ARFCN 23 since its the strongest station in the perimeter. It will nevertheless conduct power measurements on ARFCN 42 and ARFCN 61 since these are neighbours. The IMSI catcher is switched on, sending also on ARFCN 42. When the MS does its next power measurement on this ARFCN it will notice that the reception changed from -95 dB to -52 dB which is even better than the reception of the station it is currently connected to. Therefore, it will change the cell to the catcher's. Since the catcher broadcasts a different LAC, the MS announces itself by sending a Location Update.

This method will not work when a call is in progress. In that case, the only way to immediately disconnect the subscriber from the BTS and force normal cell selection mode is by jamming the frequency that belongs to the BTS.

It is important to note that from these three approaches of luring a MS to connect to a fake base station, two types of attack configurations for the IMSI catcher can be distinguished. To mimic a cell of a certain provider, the IMSI catcher has either to open up a cell with a new CID or to replace a cell. In case of opening up a new cell, the IMSI catcher has to choose a consistent configuration that blends into the environment of the respective provider, while in case of replacing a cell, the whole configuration has to be copied as to not raise any suspicion. This fundamental distinction of IMSI catcher configurations will be of help later when trying to uncover these devices.

### Risks and Irregularities

An IMSI catcher cannot target an individual subscriber, it always targets an area thus breaching the privacy of uninvolved subjects. Apart from that, a catcher that does not relay calls takes away the possibility for all connected people in the area to initiate calls.

Even if the the catcher routes calls into the network, it only has one SIM card and thus can only route a single call. This can be very dangerous because no emergency calls can be submitted in that area during the time of operation which can be as long as five to ten minutes [13] when used by authorities.

Another irregularity apart from using no encryption is that people caught in this area cannot be reached on their mobile phones, for they are not registered on the main network. As a consequence of the proxy functionality of the IMSI catcher, when a call is routed into the network the recipient can only see the number the catcher is registered with or 'Number Withheld', however, not the original number.

### 2.4.2. Law Situation in Germany

First reports of an IMSI catcher used by authorities in Germany dates back to 1997. Until November 2001, 35 cases of use were officially confirmed by the Bundesministerium des Inneren (BMI) [13]. It was used to fight organised and serious crime, like hostage-takings or drug traffic by the Bundeskriminalamt (BKA) and Bundesgrenzschutz (BGS). Attempts have been made by the government, to move the catcher out of the legal grey zone and use the *GA 900* with its capabilities of tapping in to calls for crime prosecution. At that time however the attempt was dismissed.

On 14$^{th}$ of August 2002, with Section §100i of the Strafprozessordnung (Code of Criminal Procedure), a law basis was given to the device. Afterwards, on 22$^{nd}$ of August 2006 this section and its accordance with the Grundgesetz (Constitution) was affirmed. The use of an IMSI catcher with prior authorisation by a judge does not affect peoples' right to privacy, nor does it contradict the Datenschutzbestimmungen (Secrecy of Confidential Data) or the Fernmeldegeheimnis (Secrecy of Confidential Communication). In Austria the need for a prior authorisation by a judge was removed in January 2008. During the first four months of 2008, 3800 cases of catcher use were reported in Austria [30].

Gradually, starting with §100i it has become easier for the police and agencies to use electronic surveillance. Although in 2004, it was decided by the Federal Court of Saxony that electronic surveillance is not to be used in the substantially intimate sphere of private premises. This regulation can be overthrown, if linked to the field of serious crimes and terrorism. Section §100a(1) describes that the police merely needs to show certain evidence, underpinning a suspicion that a criminal act was committed. This threshold can often be overcome easily, since it is hard for courts to check evidence for sufficiency thoroughly given the short time frame of response [22].

In contrast, the law situation considering non-authorative use in Germany is clearly laid out. The law is breached in several points when an individual operates a chatcher. One breach is sending on frequency bands that are registered to different providers thus interfereng with regular communications. However, it is very hard to prove in retrospect that an IMSI catcher has been operated in a particular area. The easiness of obtaining a self-built device and the fact that illegal operation is near impossible to prove shows the immediate risk that comes from these devices.

# 3. IMSI Catcher Detection System

This chapter will give an outline if the ICDS, the technologies and techniques used. The first part summarises the frameworks and hardware upon which the system has been developed. From this point on, the second part explains how this framework can be used to harvest information and describes the process that is used by the ICDS to evaluate this information. The last part shows how to configure and use the system to gather information from the surroundings and unveil IMSI catchers.

## 3.1. Framework and Hardware

The following section will give an overview of the OsmocomBB framework and how it works in conjunction with the Motorola C123 mobile phone to enable information harvesting for the ICDS. OsmocomBB is one of many Open source mobile communications (Osmocom) projects[1]. It delivers an open source implementation for the base band chip for certain mobile phones. Another Osmocom project is OpenBTS which delivers software for configuring and operating a BTS. OpenBTS was used to realise the open source IMSI Catcher [27] and the base station that will be used later to evaluate the performance of the ICDS.

### 3.1.1. OsmocomBB

OscmocomBB implements the baseband part of GSM as an open source project. Baseband part in this case means that it is an open source software to control the baseband chip inside the mobile phone. The baseband chip is the processor which manages the radio functionality of a mobile device. The goal is to have a phone, when using compatible hardware, operating on open source software only, as opposed to proprietary baseband implementations. Therefore, the project scope is implementing GSM Layer 1–3 as well as hardware drivers for the baseband chipset. A simple user interface on the phone is planned but not yet implemented. At this stage a verbose user interface on the computer is used. The implementation being open source is beneficial to multiple areas [20]:

- **Security:** The software running on the baseband chips is highly proprietary and closed. The source is often disclosed only to the mobile phone manufacturers using

[1] Osmocom, `http://osmocom.org/` [Online; Accessed 04.2012]

the specific chipset. One cannot be sure that this software does not have bugs that could be exploited and ultimately pose a security risk to the subscriber.

- **Education:** Currently knowledge about GSM and its layers on a technical level is not very well spread. An open source implementation as a reference could serve to educate more developers, generally interested in the subject of mobile communications and thus improve products and software. Additionally, this implementation enables universities to hold practical lab courses and interested individuals to do hands-on experiments.

- **Research:** A free implementation can decouple research on GSM technologies from the industry, because key technologies are no longer only available to researchers employed by a specific company. Additionally, security flaws can be uncovered and fixed more easily. Modifications to the protocol stack can be deployed and tested in a real environment. It is also possible to redirect all received and sent packages directly to Wireshark[1] for further analysis.

### Project Status

At this point, Layer 2 and Layer 3 do not actually run on the phone but rather on a computer to which the phone is connected via a serial cable. Layer 1 runs inside the custom firmware on the ME itself, since the procedures involving Layer 1 are very time critical. This has advantages as well as disadvantages. The disadvantage is that in order to run an application written using OsmocomBB you always have to have a computer in addition to the phone. The benefit, however, is that during the development process, the phone does not have to be touched after an initial deployment of the firmware. This means code can be modified, compiled and tested locally without the need of remote debugging. Experimenting is considerably easier this way. This separation would not work in the original GSM specification. This is why an extra interface layer between Layer 1 and 2 had to be implemented to handle messaging over the serial interface between the two original layers. It is called Layer 1 Control, L1CTL.

The current state of the project is, according to a presentation given on the 27$^{th}$ Chaos Communication Congress[2] by Dieter Spaar and Harald Welte that the network Layers 1–3 are fully implemented, SIM cards can be accessed or emulated and GSM cell selection and reselection are working. A3/A8 as well as A5/1 and A/52, Full Rate and Enhanced Full Rate codecs are there, so it is possible to do voice calls with an OsmocomBB application written for that purpose, called `mobile`. It features a terminal / telnet based interface, much like Cisco routers, however, there is no user interface for the phone so far.

[1] Wireshark, `http://www.wireshark.org/` [Online; Accessed 04.2012]

[2] 27C3 public wiki (Day 3), `http://events.ccc.de/congress/2010/wiki/Welcome` [Online; Accessed 04.2012]

| Component | Specification |
|---|---|
| Band | GSM 900, GSM 1800 |
| Size | $101 \times 45 \times 21$ mm |
| Weight | 86 g |
| Battery | 920mAh Li-Ion battery |
| Digital Baseband | Texas Instruments Calypso |
| Analog Basenand | Texas Instruments Iota TWL3025 |
| GSM Transceiver | Texas Instruments Rita TRF6151C |

Table 3.1.: Technical specifications for the Motorola C123.

### 3.1.2. Motorola C123

Since the general idea behind OsmocomBB was to become a vendor independent open source GSM implementation for everyone to use, there were certain requirements, the targeted hardware would have to meet. For the consumer side requirements, these were having a low price and a good availability. This criterion rules out do-it-yourself (DIY) approaches since the number of produced devices would be low and thus costly or a significant amount of technical knowledge would be expected from all users to assemble the hardware. For the developer side, this would also mean implementing a lot on the lower levels of analog logic. Therefore the Motorola C123 was chosen, an old, very cheap phone that is well spread. It has the advantage of being very simple on the hardware side and very well documented because the technical documentation for the Texas Instruments Calypso Chipset [28] has been leaked. The TI Calypso is the baseband chipset that is used by the Motorola C123. Table 3.1 shows an overview of the main specifications for the phone. The OsmocomBB framework should work well or with small adjustments for any phone that is based on the same components. Figure 3.1 shows an image of the Motorola C123 circuit board with the components mentioned before. Another reason for choosing this hardware platform was that during the start of the OsmocomBB project, an open source implementation of GSM Layer 1 was already available on Sourceforge (TSM30 Project) that could be used as a reference. At this point the original project has been removed from the Sourceforge site.

In order to use the Motorola C123 in combination with the OsmocomBB framework, the custom firmware implementing Layer 1 and L1CTL has to be flashed onto the board. This has to be done using a RS332 serial cable that is connected to the 2.5 mm audio jack. The audio jack of the Motorola C123 and other Calypso based mobile phones typically have a 3.3 V serial port on their audio jacks. These cables are normally referred to as T191 unlock cables. A variety of stores around the internet sell the cables ready made for about $10–$15. One must be careful when using the PC's serial port to communicate with the phone though. Since the phone's serial operates at 3.3 V and is internally connected

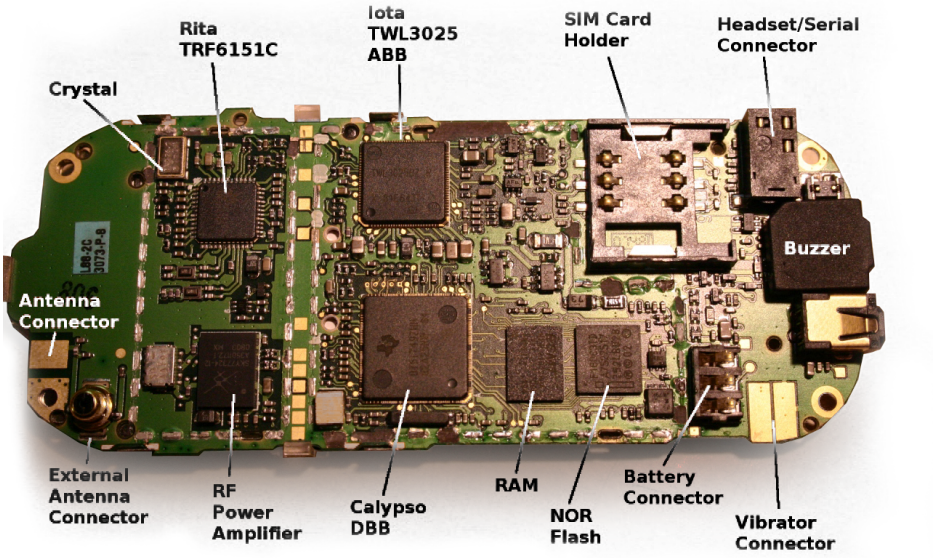



Figure 3.1.: Circuit board of the Motorola C123 with its components [19].

to the 2.8 V IO-pins of the baseband processor, directly connecting it to the computer's 12 V serial port will destroy the hardware. Therefore it is recommended to use a USB serial cable. Schematics for such an unlock cable are given in Appendix B.3.

### 3.1.3. OsmocomBB and ICDS

The setup that is used for the ICDS project can be seen in Figure 3.2. It was built and tested in a Xubuntu 11.10 environment[1] which is a more lightweight variant of the popular Debian based Ubuntu Linux distribution. The process of acquiring, compiling and running the OsmocomBB framework in this environment is explained in detail in Appendix B.1.

When setting up the system, it is recommended *not* to use a virtual machine. The bootloader and the firmware can fail to be deployed correctly if a virtual machine is used as development system. This is because the protocol used by Motorola to do the actual flashing process is *very* time critical and thus timeouts can occur that are caused by the overhead the virtual machine imposes on the hardware / software communication.

As can be seen in Figure 3.2, Layer 1 of the OsmocomBB GSM stack runs on the phone which is connected via a serial cable to the computer running the ICDS. On the computer side the `osmocon` program provides a general interface to the phone. `Osmocon` is also used to load the firmware up to the Motorola C123. Other software can communicate

[1] Xubuntu, `http://xubuntu.org/` [Online; Accessed 04.2012]

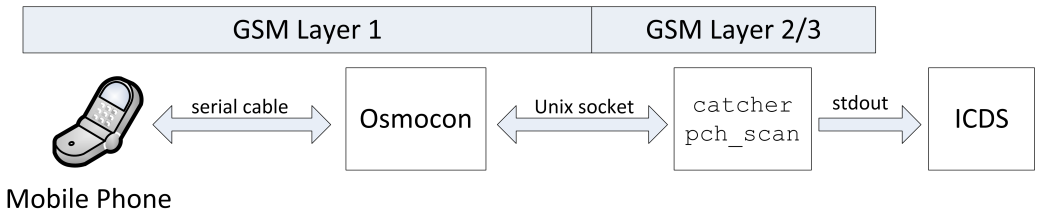



Figure 3.2.: Interaction of the OsmocomBB components with the ICDS software.

with `osmocon` and subsequently with the phone using Unix sockets.

The program `catcher`, the OsmocomBB part of the ICDS, is a modified version of `cell_log` by Andreas Eversberg that interfaces with `osmocon` to harvest information from BTSs and forward it to the core ICDS. It can be seen as a Layer 3 program that scans through available frequencies and reads information from the BCCH whenever one such channel is available on the frequency at hand. The forwarding is done directly via `stdout` since it runs as a child process of the ICDS. In a similar way, `pch_scan` gathers information on the PCH of a specific base station. The functionality of `catcher` and `pch_scan` will be explained in detail in Section 3.2.1 while the implementation and operation of the ICDS will be discussed in Section 3.3.

## 3.2. Procedure

The main goal of the ICDS is to reach a conclusion on whether it is safe to initiate a phone call or not, in other words if the base station our mobile phone will connect to is trustworthy. As mentioned before, as soon as a subscriber connects to an IMSI catcher, information on his / her location is automatically given up. Therefore, this project will use a passive approach on information harvesting, meaning we will only use information that is broadcasted or freely available as to not give up any hints of the ICDS being active.

To that end a four-step process is taken. First *information is gathered*. This process is explained in detail in Section 3.2.1. After information on the surrounding BTSs is ready inside the ICDS, a set of checks is evaluated on each base station individually, with each yielding a specific result for the station. These checks are called *rules* and discussed further along with the next two steps in Section 3.2.2. Afterwards, the results the rules yielded for each base station have to be aggregated into one single result for each BTS by an *evaluator*. At last, after every BTS has its evaluation, it can be decided whether to *tell the subscriber* if it is safe to initiate a phone call or not.

| TC | System Information Type |
|---|---|
| 0 | Type 1 |
| 1 | Type 2 |
| 2,6 | Type 3 |
| 3,7 | Type 4 |
| 4,5 | Any (optional) |

Table 3.2.: Type Codes and the corresponding System Information Types [10].

### 3.2.1. Information Gathering

As explained in Section 2.3.2, every base station has an associated BCCH where information about the station and its network is spread. BCCH frames are always sent inside a 51-Multiframe. After the MS has synchronised using the values on the FCCH and SCH, it can determine which kind of information is hosted inside the BCCH message. These so called *System Information Messages* originate at the BSC and are produced for each BTS individually and then periodically broadcasted. Since all the required information would not fit inside a single frame, there are different kinds of System Information Messages that are distinguished by their Type Code (TC) and host different kinds of information. The type can be extracted using the Frame Number (FN) of the frame the message is sent in [10]:

$$\text{TC} = (\text{FN div } 51) \bmod 8$$

Table 3.2 shows how the TCs can be mapped on those types. For this project the System Information Type 1–4 are of interest because these are available to all MSs that tune in to the particular BCCH of the respective BTS, without actively connecting to it.

The information contained inside the System Information Messages is harvested via the `catcher` program. `Catcher` is implemented inside the OsmocomBB framework and connects over the `osmocon` application to the Motorola C123. At first, a sweep scan is done over all the ARFCNs to measure their reception levels, in order to determine where base stations and thus BCCHs are located. Afterwards `catcher` tunes the phone to those specific frequencies where a BTS was found.

At each such frequency, it waits until all the System Information Messages are gathered and extracts parameters where possible. The parameters, along with the raw data are forwarded to the main ICDS application for further evaluation. An example of a fully parsed System Information Type 2 Message can be seen in Figure 3.3 [17]. The Neighbouring Cell List for example which is a very valuable source of information, is located in inside the highlighted section of the message. Examples for all the System Information Messages used, along with an interpretation are located in Appendix D and information on how they are interpreted can be found in 3GPP TS 44.018 [6]. As long as scanning mode is active, all the available stations are scanned repeatedly and changes in the BTSs will

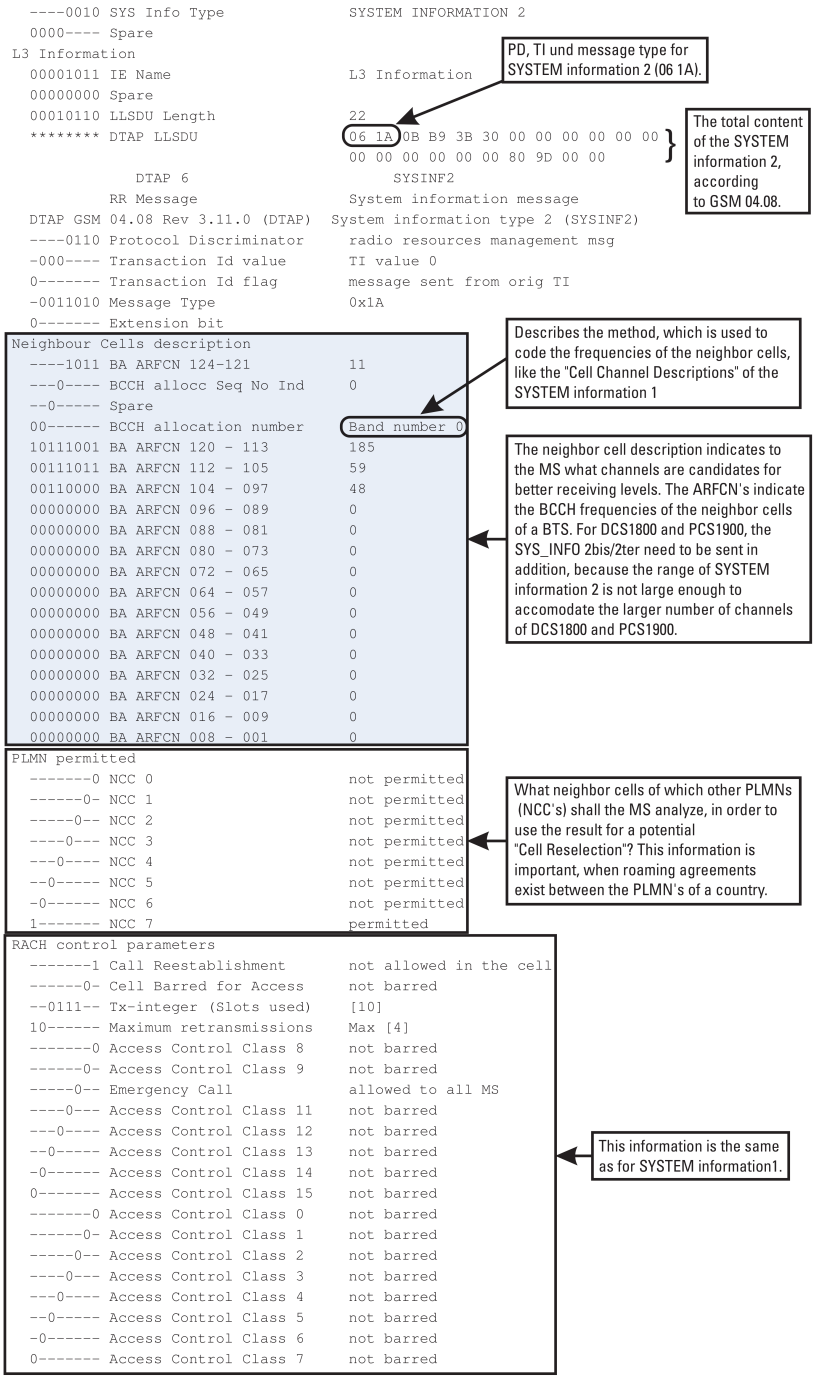



Figure 3.3.: System Information 2 Message [17].

continuously update the data model inside the ICDS software. The parameters currently harvested are:

- MCC: The Mobile Country Code the base station is broadcasting.
- MNC: The Mobile Network Code the base station is broadcasting.
- ARFCN: The ARFCN on which the base station is located.
- rxlev: Receiving strength in dB. This parameter is measured by the Motorola C123 and not part of the System Information Messages. Even small changes in the location can have a large impact on this parameter due to shadowing and reflection.
- BSIC: Because of frequency reuse in a cellular network, it is possible that two different base stations can send at the same ARFCN. In order for the MS to keep these apart a Base Station Identification Code (BSIC) is broadcasted by each BTS. It consists of a Network Color Code (NCC) identifying the provider, so the MS can filter out messages that it does not need beforehand and the Base Station Color Code (BCC) that must be unique for a given provider over all base station in a large area.
- LAC: This is the last part of the LAI (that consists of MCC + MNC + LAC) and is a hierarchical identifier for a given set of base station. The hierarchy is provider wide, meaning two different providers may use LACs with a completely different numbering system. The LAC is used by the provider to tell the MS that it entered a new area and has to announce itself.
- CID: The CID is a unique identifier for the cell the MS is connected to. Unique in this case means unique in a large area so that a mobile phone should never receive the same CID for different base stations.
- Neighbouring Cell List: Each base station keeps a list of other base stations in the perimeter for the MS to scan and determine if there is a BTS with a better reception in the area.

Note that there are different formats for the Neighbouring Cell List since the original number of 17 bytes could only present a bit mask for 124 neighbouring ARFCNs. This works for the 900 MHz band, but for the extended 900 MHz and the 1800 MHz band the System Information Type 2bis and System Information Type 2ter have to be harvested additionally, to construct the Neighbouring Cell List.

The `pch_scan` tool does not rely on the BCCH but rather on information available on the PCH, as the name implies. If a mobile phone is connected to a base station and not actively participating in a communication process, it is in a passive mode to save battery, waiting for either the user to initiate communication or the network to contact it.

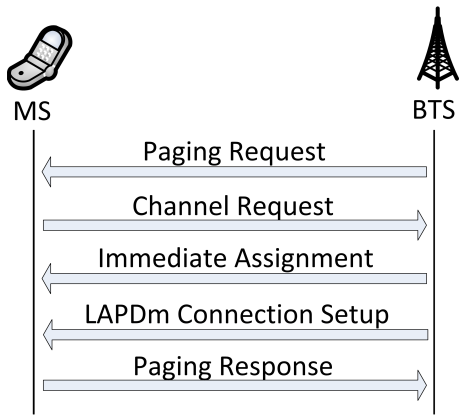



Figure 3.4.: Procedure taken when the network has a call / text waiting for a passive subscriber.

As mentioned in Section 2.3.2, the network contacts the MS on the PCH if there is a text message or a call, waiting to be delivered. The procedure is outlined in Figure 3.4. A paging request by the network is answered by the MS by requesting a dedicated channel, which is assigned by the network in turn with an IA. From this point on the connection can be set up.

The `pch_scan` listens for activity on this channel and harvests the following information:

- Paging Messages: The ICDS is informed about every Paging Message that has been caught.
- Immediate Assignment: If an IA is caught, it is logged and parsed. The TMSI to which the IA was sent as well as the assigned channel number and whether it is a frequency hopping channel or not is forwarded to the ICDS.

### 3.2.2. Information Evaluation

Each base station is evaluated, the moment the data completely arrived at the ICDS application. Additionally, when a new BTS has been found and added, all formerly discovered stations are also re-evaluated since new discoveries can have an impact on the rules that evaluate the context surrounding an old base station.

As mentioned above, evaluation is done based on constructs called *rules*. Each rule represents one check that can be performed on a base station and yields a result based on its findings. A rule can also be seen as a mapping from a set of input parameters to one of the values *Ok*, *Warning*, *Critical*, *Ignore*.

$$\{\text{Base station parameters}\} \mapsto \{\text{Ok}|\text{Warning}|\text{Critical}|\text{Ignore}\}$$

| Rule | Functionality |
|---|---|
| Provider Known | Checks whether the provider is in a list of known providers. |
| Country / Provider Map | Checks whether the given provider is a valid provider for the given country. |
| LAC / Provider Map | Checks whether the LAC of the station is in the normal LAC range for that provider, given the area. |
| ARFCN / Provider Map | Checks whether the ARFCN is in the officially registered range of the provider. |

Table 3.3.: Configuration Rules implemented inside the ICDS.

A *Critical* result means that the base station evaluated has a critical configuration error or critical settings that are not found on normal base stations, e.g. unknown provider names or empty neighbourhood lists. This station should not be trusted.

If a *Warning* status is yielded, the BTS at hand has some concerning features but it could not be said whether it really is an IMSI catcher or sheer coincidence. An example would be a base station having a Neighbouring Cell List of which none of the cells therein have actually been discovered. The list could either be a fake or it could simply be coincidence that the scan has not found any. They could have been out of range for example.

In some cases a rule cannot yield a finding. That is when the state is explicitly set to *Ignore*, so the evaluator knows that this rule should have no influence on the final outcome. This is the case for example when a rule refers to a parameter that has not been looked up or scanned.

If everything went as expected, *Ok* is returned.

The rules can be divided into four different categories depending on how they work and which situations they are tailored to. Most of the rules are parametrised, so they can be tweaked to different environments and standards. The different rule categories are *Configuration Rules*, *Context Rules*, *Databse Rules* and *Scan Rules*.

### Configuration Rules

The first set of rules, called *Configuration Rules*, targets the base station itself. Rules in this category are meant to check parameters of a single BTS for integrity and configuration mistakes that could have been made by an IMSI catcher operator. An overview of the Configuration Rules that are currently implemented inside the ICDS is given in Table 3.3.

A few things have to be noted when configuring these rules. Since there is no official listing or rule on how the LAC should look like, the LAC / Provider Mapping Rule needs

| Rule | Functionality |
|---|---|
| LAC Median Deviation | Checks whether the LAC of the given BTS deviates more than a certain threshold from the median LAC of that provider. |
| Pure Neighbourhoods | Checks whether all stations found in the Neighbouring Cell List share the same provider. |
| Neighbourhood Structure | Checks the structure of the Neighbouring Cell List for certain patterns. |
| Discovered Neighbours. | Checks whether a certain amount of the cells in the Neighbouring Cell List have actually been found. |
| Cell ID Uniqueness | Checks whether there are other cells with the same CID. |

Table 3.4.: Context Rules implemented inside the ICDS.

knowledge of the area in which the ICDS is used. The ICDS itself can be used to gather that knowledge, but it has to be done prior to using the rule for base station evaluation. The ARFCN range each provider has registered in Germany can be looked up at the website of the Bundesnetzagentur[1] which is needed for the ARFCN / Provider Mapping Rule.

The main problem at this point is that all the parameters that can be checked by these rules can also be set by the operator of the IMSI catcher. If these are set in a consistent way, this set of rules is not sufficient to identify a catcher. Therefore, another set of rules has to be added that incorporates information of surrounding nodes.

### Context Rules

The second set of rules is called *Context Rules*. As the name suggests, these rules serve the purpose of checking how well a given BTS fits into its neighbourhood. Table 3.4 shows which rules have been implemented.

For the LAC Median Deviation Rule, the median was chosen over the average since an extreme value (ill configured IMSI catcher) would have too strong an impact on the average, to which all the BTS are compared. It could even have such a strong effect on the average that legitimate base stations would fall below the threshold and be recognised as catchers. The threshold, at which deviation a node is evaluated as being *Critical* can

[1] Bundesnetzagentur Vergabeverfahren,
`http://www.bundesnetzagentur.de/cln_1911/DE/Sachgebiete/Telekommunikation/RegulierungTelekommunikation/Frequenzordnung/OeffentlicherMobilfunk/VergabeVerfahrenDrahtlosNetzzugang/vergabeVerfahrenDrahtlosNetzzugang_node.html` [Online, Accessed 04.2012]

be set in the configuration section for the rule. A value of 0 would mean that no deviation from the median is allowed. This could lead to problems as some experimental scans have shown. However, in none of the scans more than two different LAs have been found per provider and since these were neighbouring areas, the difference in the code was only 1. For the Freiburg area a 1% threshold for the deviation yielded good results.

**Neighbourhood Structure** The Neighbourhood Structure is the graph that is described by the Neighbouring Cell List located in the System Inforamtion 2 / 2bis / 2ter constructs. Figure 3.5 shows an extract of the neighbourhood graphs at the Faculty of Engineering of the University of Freiburg[1]. The E-Plus subgraph has been enlarged. It can be seen that for each provider, the neighbourhood forms an isolated, nearly fully connected subgraph. Nodes with a green background have an *Ok* rating, while the red node has a *Critical* rating. The bordering white nodes have not yet been discovered and evaluated, therefore, they have no outgoing edges, for no Neighbouring Cell Lists have been extracted. They were merely found by extracting the Neighbouring Cell Lists of other nodes. This could be the case because they are too far away for the Motorola to receive or because of signal damping due to shadowing and reflection effects. In the ICDS, the aspect of isolated subgraphs for neighbourhoods is captured inside the *Pure Neighbourhoods Rule*.

An interesting fact is that one node inside the E-Plus subgraph on the upper right is marked *Critical*. This is because it is a BTS of the university's own GSM network. It was set up to be in a E-Plus neighbourhood but is not consistent with the E-Plus nodes surrounding it. Therefore it is marked by the ICDS.

The node was set up inside the E-Plus neighbourhood for another Master Thesis [32] at the Chair of Communication Systems where the goal was to estimate the most probably position of a subscriber, given his / her reception levels.

Some of the attacks discussed in Section 2.4.1 imply a certain structure of the neighbourhood graph. Since the IMSI catcher tries to lock in MSs that have connected from switching back to a normal cell, the neighbourhood list of such a catcher cell would either be empty or would only host neighbour cells that have a lower reception strength than itself.

An empty Neighbouring Cell List is represented in the graph by a node that has been discovered and has no outgoing edges. A Neighbouring Cell List containing only imaginary nodes serves the same purpose. Figure 3.6 shows a simplified, regular neighbourhood graph, compared to a graph with two catcher nodes inside. In this case, catcher C chose the attack where it replaces a previously existent BTS, whereas catcher D opened up a new cell. Replacing has several advantages, one being already integrated in the neighbourhood of other nodes. Mobile phones will constantly monitor the reception strength of all neighbouring nodes and thus also the reception strength of the IMSI catcher which replaced one. For catcher D it is the other way around, it has only outgoing edges. This

[1] Georges Köhler Allee, Freiburg

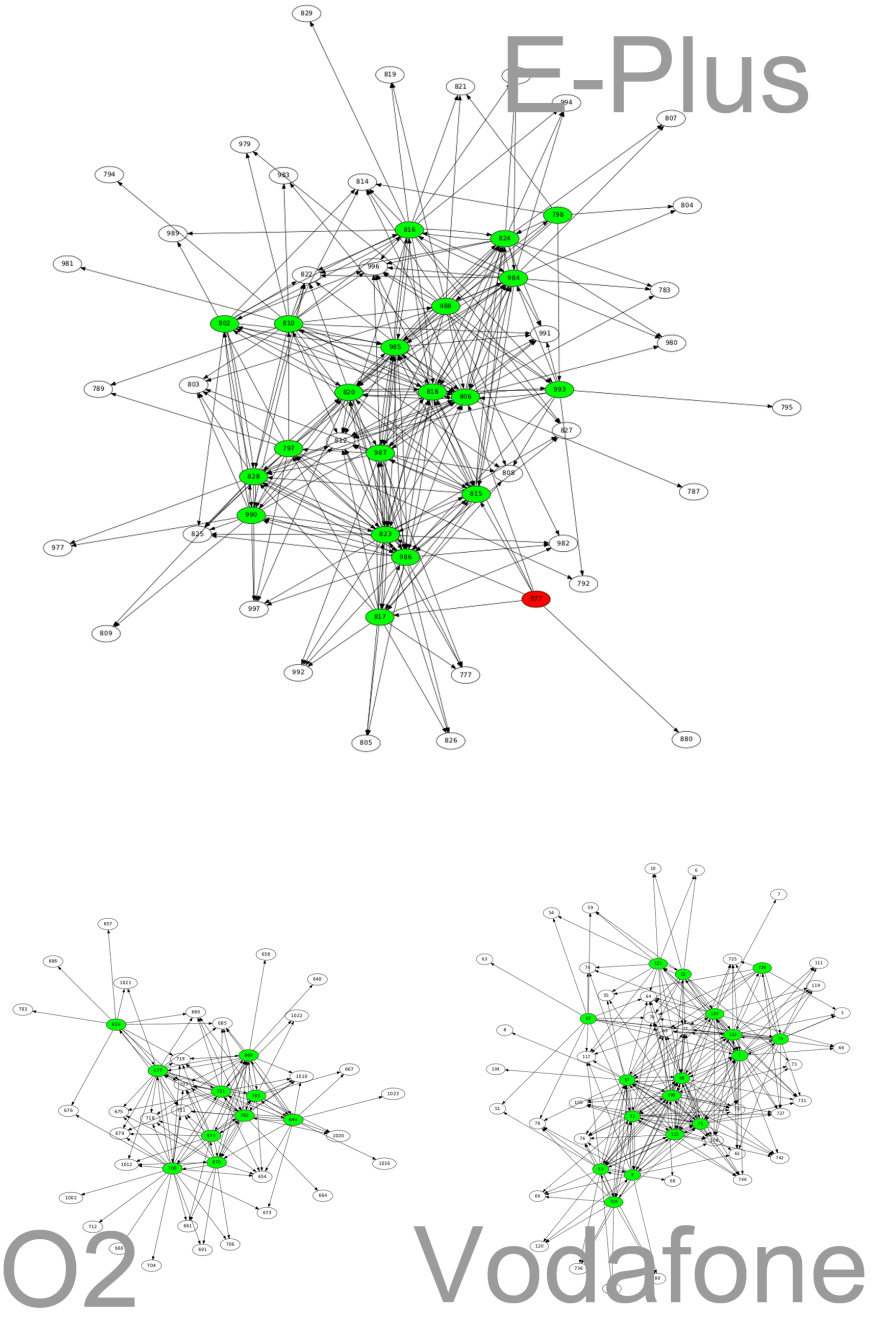
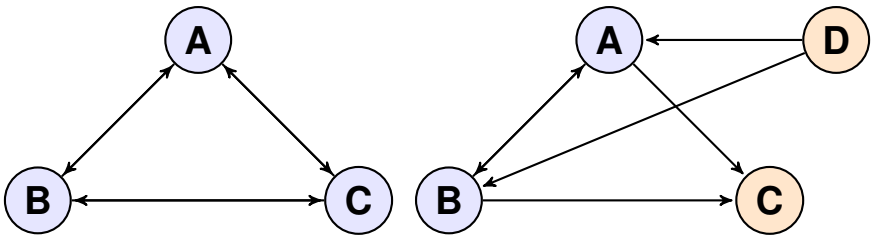



Figure 3.5.: Some base stations and their neighbourhood connections at the Faculty of Engineering.




(a) Normal neighbourhood (b) Tainted neighbourhood

Figure 3.6.: Comparison between a normal neighbourhood subgraph and a tainted one.

means that this cell is not known by any other node of the same provider. Nevertheless, it has some outgoing edges to nodes with significantly less transmission strength to not stick out too much as a completely isolated node. Combinations of these two approaches are also possible. These thoughts are basically what is captured inside the *Neighbourhood Structure Rule*. The procedure the Neighbourhood Structure Rule follows is:

1. Check if the node in question has neighbours and check if at least one neighbour has been discovered. This rules out the cases where IMSI catchers have no neighbours or only an imaginary list.

2. If no neighbours have been discovered by the ICDS, check if other nodes share some of the neighbours, if yes yield a *Warning*, else yield *Critical*. If the node is question is a legitimate node and the rare case occurs that none of its neighbours are in reach, most of its neighbours should be shared by other nodes of the same provider.

3. Check if other nodes of the same provider have the node in question inside their neighbourhood list, e.g. if the node in question has incoming edges. This would not be the case, for example, for an IMSI catcher that broadcasts on a new ARFCN.

4. If none of the above criteria suggested otherwise, yield *Ok*.

This rule cannot find an IMSI catcher that has in- and outgoing edges, in other words a device that replaced a legitimate base station and copied the Neighbouring Cell List from the original cell. Such a catcher would transmit at a very high strength and thus make sure all its neighbours have a worse reception on the target mobile phone. It is generally not possible, to rule out base stations where all outgoing edges point to base stations with a lower reception, since every legitimate neighbourhood will have one node that excels all other nodes in terms of reception.

The Neighbourhood Structure Rule tests if at least one neighbour has actually be found. To raise this threshold the *Discovered Neighbours Rule* can be used. It takes a parameter as an input which is interpreted differently depending on its range. If the threshold is

| Rule | Functionality |
|---|---|
| Cell ID Database | Checks all CIDs in the area against a database. |
| Local Area Databse | Checks whether the LAC of the given BTS deviates. |

Table 3.5.: Database Rules implemented inside the ICDS.

in the interval $[0, 1]$ it is interpreted as a percentage. $0.5$ meaning that at least half the neighbours in the list need to be found for the rule to give an *Ok* rating. A threshold in the interval $(1, +\infty)$ means that this absolute number of base stations have to be found. If a floating point number is provided the decimal places are stripped. As an example $3$ and $3.84$ would both mean that at least $3$ neighbours would have to be found. This representation cannot cover the 'at least one' statement since $1$ equals $100\%$, which is no problem for this case is already covered by the Neighbourhood Structure Rule.

### Database Rules

Let us do a quick summary of the situation so far. To investigate the current possibilities unveiling a catcher, we will look over the parameters with the two attack types presented in Section 2.4.1 in mind. For both attack types presentedm it is possible to find a parameter configuration that does not raise suspicion, if the operator chooses a compatible ARFCN, etc. for the mimicked provider. Therefore the Configuration Rules and most of the Context Rules will yield an *Ok* result.

The Neighbouring Cell List is a bit different. Since the catcher wants to keep lured subscribers, it will normally have an empty list or a list pointing only to BTSs imaginary neighbours. Both of these cases can be detected. However the operator *may* also choose to set a list consistent with the neighbouring cells, e.g. a catcher replacing a cell and copying the neighbourhood list.

Another parameter has to be introduced to yield information in the cases the rules mentioned before fail, the CID. For the CID, there are basically two possibilities depending on which attack type is used. The first possibility is that the IMSI catcher opens up a new cell and the second one is that it replaces a formerly existent cell. In the first case parameters can be chosen in a consistent way although a new CID has to be chosen, as the CID needs to be unique. In the second case all parameters can be copied from the original cell. Both possibilities can be resolved by adding outside knowledge to the ICDS thus circumventing the problem of other parameters being forged. This is done by rules called *Database Rules*.

Table 3.5 shows the rules that each handles one of these cases. The first case is the easier of both. We know that the catcher cell has a new CID that has not been there before. Therefore, the *Cell ID Database Rule* has two different means to exploit this fact:

| Rule | Functionality |
|---|---|
| rx Change | Watches out for changes in reception. |
| LAC Change | Watches out for changes in LACs. |

Table 3.6.: Scan Rules implemented inside the ICDS.

- A database of CIDs can be learned by the ICDS beforehand. This can be used to detect new CIDs that have not been seen before.

- A commercial or public CID database can be used to compare against the CIDs found by the ICDS. A web service is also offered by most providers of Cell ID databases.

The three largest CID databases are the two commercial ones by Ericson[1] and combain[2] as well as the free alternative OpenCellID[3] [29]. Ericson and combain have models where a subscription or a fee per request must be paid. Another free alternative with a large coverage is Google Mobile Maps that also offers a web service where the CIDs and the respective LACs can be checked against their database to obtain localisation information (or simply check if they are contained in the database). By adding this information new cells can be identified.

The second attack type where an existing cell is replaced, is a bit more complicated since its parameters are an exact copy of the old cell. Attacking by replacing a cell works in a way that the cell with the worst reception is targeted. That way when the IMSI catcher finished replacing it, the reception goes up a significant amount and the mobile phone will move over to that cell. The difference in reception can be used to identify this kind of attack. In general the reception cannot be used well as a parameter because shadowing and reflection can substantially change the reception from one moment to the other when minimal movements have occurred. However, if reception intervals are logged for a fixed location, like an office, important calls made from that specific location can be protected against this kind of attack. To that end, the ICDS can monitor reception levels to build up databases, with information on the reception intervals of the cells in different, fixed locations. The *Local Area Database Rule* then checks if reception levels differ significantly for a given location.

### Scan Rules

At this stage, if local information is present, an IMSI catcher should be identified with a high probability. However, if local information has not been gathered in advance, the main idea of Database Rules can still be applied. In contrast to the other three categories of rules mentioned before, *Scan Rules* evaluate parameter changes over time. This means parameters are being monitored over the duration of one or multiple sweep scans and changes are noted. The *rx Change Rule* builds upon the same idea as the *Local Area Database Rule*, only applied to a scan-to-scan basis. Changes in reception are evaluated against the last known reception level for each base station.

When watching for parameter changes, the LAC is another interesting parameter. If a mobile phone connects to an IMSI catcher due to its better reception level, the mobile phone will not immediately announce itself, thus the IMSI catcher has no knowledge that a new subscriber connected to it. A mobile phone announces itself by sending Location Updates to the network, this is only done when a certain timeout is reached or when the phone enters a new LA. Since this timeout can be very large (the lowest value possible is 6 minutes), an IMSI catcher usually sends a different LAC than the original cell, to force the MS to announce itself by sending a Location Update. IMSI catchers showing this kind of behaviour are uncovered by the *LAC Change Rule*

### Remaining Issues and Paging

If a catcher is configured in a consistent way, replaces a cell and by chance has an *appropriate transmission power*, the ICDS will not unveil it up to now, if it also chooses to maintain a non-suspicious Neighbouring Cell List and does not transmit a new LAC. *Appropriate transmission power* in this case means that the reception of the catcher does not differ significantly from the reception of the original base station.

An IMSI catcher is not part of a provider's network, it is merely a proxy for a base station. At best, it can route calls into a network but it cannot take calls that are intended for a subscriber and route them. Therefore, an IMSI catcher will not send Paging Messages to connected subscribers, while a normal base station will have a very high number of pagings depending on the number of subscribers that are connected. This is a significant difference between a catcher and a regular base station.

This is an additional information that can be used to identify an IMSI catcher. The program `pch_scan` tunes the Motorola C123 to the PCH of a particular base station and gathers Paging Messages and IAs. If no Paging Messages could be collected during a longer period of scanning it is a strong indicator towards being confronted with an

[1] Ericson Labs, `https://labs.ericsson.com/apis/mobile-location/` [Online; Accessed 04.2012]

[2] Mobile Positioning Solutions, `http://location-api.com/` [Online; Accessed 04.2012]

[3] OpenCellID, `http://www.opencellid.org/` [Online; Accessed 04.2012]

IMSI catcher. Additionally, when IAs are found the scan extracts whether the assigned channel is a frequency hopping channel or not. Since frequency hopping is considered a security feature by providers, all German providers always assign frequency hopping channels. An IMSI catcher however may not support hopping since it does not have multiple frequencies at hand.

The *PCH Scan* feature has not been implemented as a regular rule because each given base station needs some time to be scanned. If that would be done on a regular basis for every station that has been discovered, it would delay the whole scan by a large amount of time and the interval between re-evaluations would be very high. Therefore, it was implemented as an extra feature to be used when needed. The ICDS also uses this method on particularly filtered base stations in *User Mode* as will be explained in Section 3.3.4.

### 3.2.3. Base Station Evaluation

All the rules are evaluated for each base station. Aggregation of these rule results into a single result is done by modules called *evaluators*. Currently there are two different evaluators implemented inside the ICDS:

- Conservative Evaluator: This is a worst-case evaluator. It iterates over all the rule findings and yields the most concerning finding as its result. By default this evaluator is enabled in the system.

- Grouped Evaluator: With this evaluator rules can be grouped together. Inside each group the result for the group is found by majority vote, whereas the final result is conservatively found by comparing all the group results.

Different kinds of evaluators can be used to tweak the whole system more to a specific environment or purpose, if specific rules are grouped together. They are meant more for experimental purpose, if the ICDS is used as a toolbox for analysing base stations, to give more freedom in use to the operator. In case of the system being used in *User Mode* or for the sole purpose of finding whether an IMSI catcher is active or not, the conservative evaluator should almost always be the evaluator of choice and tweaking should be done on the rule parameters rather than on the evaluator.

After a result has been determined for each station, all the results are again aggregated into a final result. The overall result depends on which mode the ICDS is used in. If it is used in normal mode, the final result will be a conservatively aggregated result over all the stations in the list. If the ICDS is run in *User Mode*, which is the mode an end user would use the system in, the ICDS looks up the provider the user has entered, filters out the base station with the best reception for that provider and yields its evaluation as final evaluation. This reflects the fact that a subscriber cannot choose the BTS it connects to, since the MS will always connect to the best base station available for its given provider.

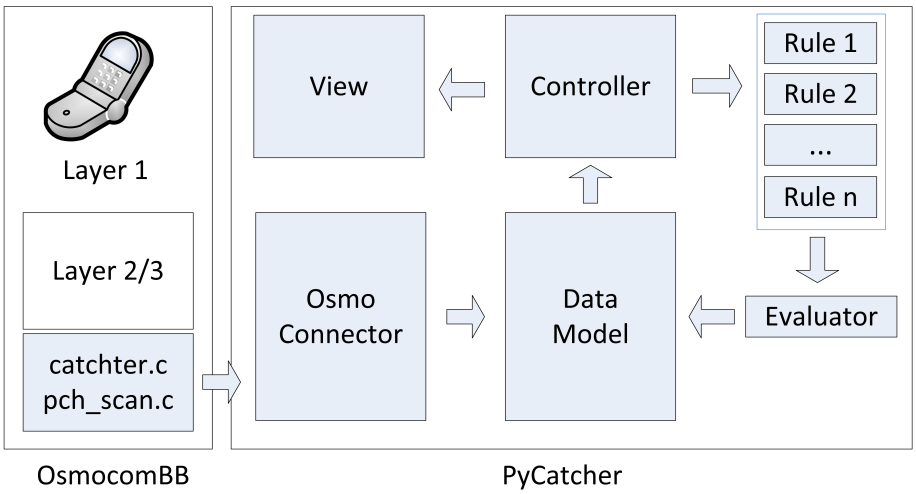



Figure 3.7.: System architecture of the ICDS. The arrows indicate the flow of data.

## 3.3. Implementation

This section will discuss some technical aspects of the ICDS software itself. The first section focuses on architectural aspects and how the architecture can be extended whereas the second and third section will then explain how to configure and operate the application.

### 3.3.1. Architecture

Figure 3.7 shows a diagram describing the system architecture, the modules in light blue have been implemented for this project. The application consists of two main parts. One part, the `catcher` and `pch_scan` programs, are implemented inside the OsmocomBB framework, the other part, *PyCatcher*, is a Python application that uses `catcher` and `pch_scan` to harvest information and evaluate it afterwards. Since the way these two sub-programs work has already been described in Section 3.2.1, this section will focus on the Python application part.

As mentioned before, Layer 1 of the GSM stack is implemented in the firmware running on the Motorola C123. Layer 2 and Layer 3 are implemented on the computer and are used by the `catcher` and the `pch_scan` software to harvest information from the BCCH and PCH respectively.

The PyCatcher application was designed with a Model View Controller (MVC) approach in mind to make it easy to implement new functionality. The MVC pattern is used to separate the data model of an application from the logic as well as from the way it is presented to the user. That way each of the different components can be exchanged

without affecting the other two. An additional module has been added, the *OsmoConnector* that is loaded by the controller and spawns `catcher` as a child process. It takes the output back in and transforms it into an object oriented representation of the discovered base stations. These are then handed over and update the data model. This way, it can be ensured that only coherent and complete information is incorporated in the data model. Another benefit is that the parsing module is isolated from the main program logic. OsmoConnector is also the module that spawns `pch_scan` when requested by the controller.

The *controller* is the main part of the program and instantiates all the other modules. It loads data from the model, triggers the evaluation and sends the results to the view to be displayed. As discussed before, there are several rules that can be evaluated for each base station. These rules are stored within the controller and can be enabled or disabled by using the view that relays new rule configurations back to the controller to be applied. Whenever a new evaluation is requested, the controller evaluates the active rules and gives the results to the active evaluator, afterwards the results are send to the view for display to the user. Note that all the structures used are view independent, this way the current view could easily be exchanged with a web interface for example.

The `view` in this project consists of a GTK3[1] window with several forms for user input. It is bound to the controller using PyGTK[2]. Details on the view and how to use it will be explained in Section 3.3.3.

Rules and evaluators were designed in a plugin fashion, since these are the main points where the program can be enhanced and new ideas can be realised. Implementing a new rule or a new evaluator works by extending the rule or evaluator base class and implementing one method inside the derived class that contains the actual logic. After that has been done, they only need to be added to the list of evaluators and rules included inside the controller. Appendix C.1 gives an example of how this can be done.

### 3.3.2. Configuration

The configuration of the system is located in the `settings.py` file. All configuration is done within the python language, where each module has its own dictionary inside which it can have an arbitrary number of parameters with their respective values or if only few parameters are required they are read in as simple variables. Figure 3.8 shows an example with the four common expressions used for parameters in this project.

The file consists of five main sections. The first one contains parameters that are needed for the correct operation of the ICDS system and have to be edited depending on the environment:

- `Device_settings`: The setting for the mobile phone that is used. In case the

[1] The GTK+ Project, `http://www.gtk.org/` [Online; Accessed 06.2012]
[2] PyGTK, `http://www.pygtk.org/` [Online; Accessed 06.2012]

```
dictionary = {
        'key_1': value_1,            #single value
        'key_2': (value_2,value_3)   #value range
        'key_3': [value_5, value_6] #list of values
}

variable = value_7               #simple variable
```

Figure 3.8.: Configuration Dictionary in the settings file.

Motorola C123 is used, this section does not need to be edited.

- `Osmocom_lib`: The path to the folder that contains the OsmocomBB framework.
- `Commands`: This is only to be edited when a newer version of the framework is used and the folder structure has changed compared to the release that was used in this project.

The second and third sections contain parameters for the different rules and evaluators. This is followed by a section to set some general parameters for the `pch_scan` tool and a section where the locations of the different databases can be changed. A completely documented configuration file with all the rule and evaluator parameters can be found in Appendix C.2. The file is read in as a python file. This way python code can also be used to change settings dynamically depending on the environment or how the ICDS is started.

### 3.3.3. Graphical User Interface

The ICDS main application has to be started with root privileges since it needs to work with Unix sockets and open up connections to the Motorola C123. This should be done by starting up the `main` class that initialises everything else.

```
sudo python /path-to-project/Src/PyCatcher/src/main.py
```

After a brief loading time the main window shown in Figure 3.9 will appear if a valid configuration is set up.

The different elements shown in the main window are:

1. Firmware Loader: This button is used to load the OsmocomBB firmware onto the Motorola C123. For this to work, the mobile phone must be connected correctly to the computer and available on the configured `tty` interface. After pressing the button, on-screen instructions will lead the user through the process of flashing.

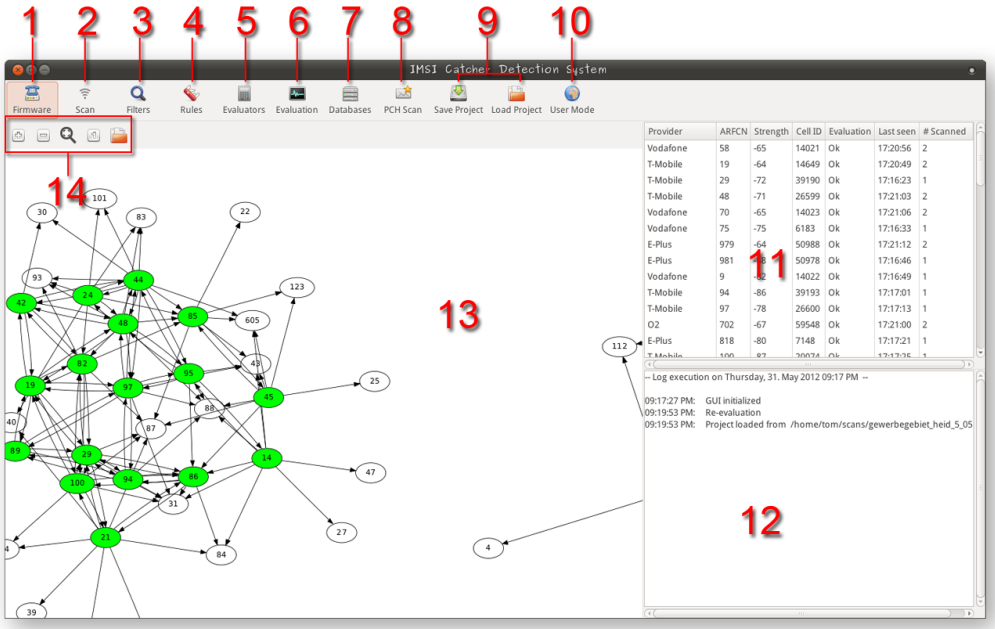



Figure 3.9.: The ICDS main window.

2. Scanner: This starts the `catcher` subprocess in the background and fills the data model with information on the discovered base stations. During this process, the Base Station List (11) and the Base Station Graph (13) will also be populated in realtime. Re-evaluation on all base stations is done for every new BTS that has been found.

3. Filter Window: This brings up the window shown in Figure 3.10(c), where different view filters for the Base Station List and the Base Station Graph can be set. Note that these filters do not modify the underlying data model or the behaviour of the scanner, they merely manipulate the view. Hidden base stations will be scanned and added to the data model independent from the filters set, so they can be viewed at a later point if necessary. Available filters are:

   - Provider Filter: Takes a comma separated white list of providers that should be shown.
   - ARFCN Filter: Takes a range of ARFCNs to be shown.

   These two filters can be combined together. Filters are designed the same way as rules and evaluators, a new filter can be implemented by derivation of the base class.

4. Rules Window: All the rules implemented inside the ICDS will be brought up with a check box to enable or disable these rules. Disabling means that they will not be considered for the evaluation of a base station. A screenshot can be seen in Figure 3.10(b). If rules are changed during a sweep scan, everything will be re-evaluated according to the new rule set, without interrupting the scan.

5. Evaluator Window: This window will let the user choose which evaluator, discussed in Section 3.2.3, to use for BTS evaluation. Choosing a new evaluator will also trigger a re-evaluation of all the data collected so far.

6. Evaluation: This button brings up a separate window, showing only the final evaluation of the scan. The final evaluation shown in this dialog *will* be affected by the filters set. Base stations that are filtered out are not considered.

7. Databases Window: The window shown in Figure 3.10(a) contains settings for all the databases the ICDS uses. These settings are mandatory if the Local Area Database Rule or the Cell ID Rule is going to be used. It is also possible to export the current scan as a Comma Separated Value (CSV) file or a Sqlite database, to be used in other programs.

8. PCH Scan Window: This button brings up the dialog illustrated in Figure 3.10(d), in which an ARFCN or a list of ARFCNs can be scanned to discover Paging Messages and IAs on the PCHs. The timeout sets the duration of a scan. Results of

the scan will be shown in a list in the lower part of the window after the scan is finished. If the checkbox is checked, the results from the scan will also carry over in the data model.

9. Save / Load Project: The current state of the application can be saved as or loaded from a `.cpf` file. This enables the user to continue a scan at a later time or to compare different data sets scanned at different points in time or locations with one another.

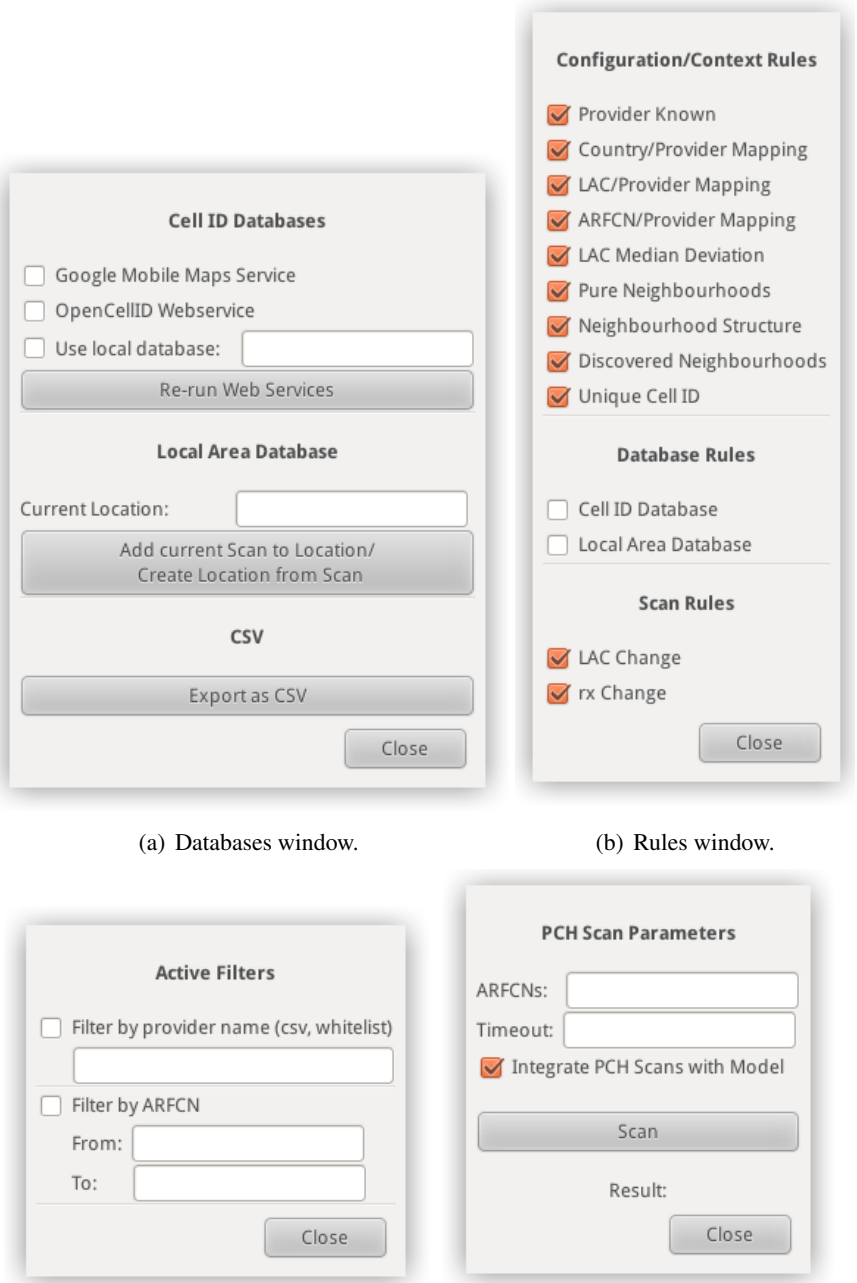
10. User Mode: The ICDS is ultimately meant to be a tool that can be used by end users to check whether it is safe to initiate a phone call or not. This dialog presents a way, the already configured system could be shown to end users. Only the provider is to be entered and a final evaluation will be returned, once the ICDS is done with the process.

11. Base Station List: This list gives an overview of which base stations have been discovered so far, along with some distinguishing information including its evaluation. A detailed view of a base station can be brought up by selecting it in the list and pressing the enter or return key. The report is separated into four main parts, the first being all the harvested parameters, followed by findings the different rules and evaluators yielded and a section with the raw uninterpreted System Information data.

12. Log Window: Every important event inside the ICDS is reported in the log together with a time stamp, when it occurred.

13. Base Station Graph: This graph displays the base stations found in the Base Station List (11). A node represents a single BTS and is labelled with its respective ARFCN. An edge from note $A$ to $B$ is drawn if node $B$ occurs in the Neighbouring Cells List of $A$. Nodes with a white background have only been found inside Neighbouring Cell Lists but not yet by the ICDS scanner itself, whereas nodes with a red, yellow or green background have been found and evaluated with the colour representing either a critical, a warning or an ok status respectively.

14. Graph Controls: These are meant to make navigating the graph a bit easier. From left to right the functionality is zoom in, zoom out, fit the whole graph to the viewport and display the graph in original size. Zooming can also be done with the mouse wheel and it is possible to drag the graph around by clicking and holding it with the mouse and then moving it in the desired direction.




(a) Databases window.

(b) Rules window.

(c) Filters window.

(d) PCH scan window.

Figure 3.10.: Dialogs for different settings.

### 3.3.4. Usage

This section will list some common use cases and explain how to setup and operate the system to achieve the desired result. Button numbering refers back to Figure 3.9.

**Conducting sweep scans:** This is the normal mode of operation, scanning and evaluating all base stations in the perimeter. This is also used for gathering various kinds of information to be used for analysis later. At first the firmware needs to be flashed onto the device by pressing (1). After the flashing process is finished, the scan can be started by pressing (2). Either before or during the scan (3),(4) and (5) can be used to customise the output or rules that should be considered during evaluation. The scan can be stopped at any time. Resuming the scan will renew the information in the Base Station List. The scan will continue renewing information until it is terminated by the user. The number of times a specific BTS has been scanned is shown in the *Sightings* column of the Base Station List.

**Using and obtaining CID Information:** CID information can be obtained through several different means. The Databases window shown in Figure 3.10(a) can be brought up by pressing (7). In the upper part, settings concerning the acquisition of CIDs can be found. The operator has the choice between three different methods which can also be used in combination. *Google Mobile Maps Service* compares the stations' CIDs and LACs to the ones in the Google database. If they are found they are marked as such and additionally their location information will be set. *OpenCellID Web Service* performs the same task if activated. As of now, OpenCellID has a very low coverage compared to Google's service but it has been included since it is an open source approach that is in development and updated constantly. The *Use Local Databse* feature allows to use a previously build Local Area Database as Cell ID Database for lookups. For this purpose the location to be used as database has to be entered in the textfield, e.g. 'office' or 'home'. Offline lookups can be done that way, which are considerably faster that online lookups. Since these lookups take some time, if performed using webservices, this is not done while the scan is taking place, to not delay the acquisition of information from new base stations. Pressing the button below the checkboxes will add the Cell ID Database information from the selected sources to all the stations currently in the base station list. If more than one service is activated, lookups will be done starting with the Google service, if active and using the next one in line only if the previous lookup failed. Having at least one service activated and run on the base station list is a precondition for the Cell ID Database Rule to work.

**Building or using a Local Area Database:** Having set up the correct location in the *Current Location* field of the databases window and having a valid database for that location are preconditions for the Local Area Database Rule to work. To build up a database


Figure 3.11.: The User Mode window.

for a specific location a sweep scan for this location has to be done. After the sweep scan is finished, the current location, e.g. 'office', has to be set in the dialog and the button for adding / updating the database has to be pressed. If there was no existing database for that location, it will be created, otherwise the database will be updated with the new information acquired by the sweep scan. To enhance the quality of a Local Area Database it is recommended to do multiple sweep scans and integrate them rather than relying on a single scan only. This raises the probability that all BTS in the perimeter are found and it solidifies the interval in which the base station signal strength varies.

**Conducting a PCH Scan:** A PCH scan can be conducted in addition to a sweep scan or as a standalone method, therefore, no scan data needs to be present. Since PHC scans and sweep scans both use the Motorola C123, a PCH scan can only be done when no sweep scan is active and vice versa. The first parameter is a comma separated list of ARFCNs that will be scanned. The second parameter is the timeout. A scan for a particular ARFCN will tune in on the PCH of each ARFCN given and wait there until the timeout is reached, gathering all Paging Messages and IAs that are sent in that time interval. In the lower part of the dialog, after the scan has finished, the statistics for the scanned BTSs will occur. If the checkbox is checked, the data acquired by the scan will also be integrated with the data model and will have an impact on the evaluation displayed in the Base Station Graph. The findings can then also be seen in the report for a base station.

**Utilising User Mode:** Data needs to be present inside the ICDS either by loading a project file for the corresponding location the system is used in or by having performed a sweep scan in advance. There is only one input field in the dialog as Figure 3.11 illustrates. The user has to enter the provider name in this field and push the *Start Evaluation* button. From the scan data, the ICDS extracts the base station with the highest reception for the given provider since this would be the station a MS would connect to if started up. If the station already has been evaluated as *Critical*, *User Mode* will instantly yield this as result. In all other cases it performs an additional PCH scan on that station to rule out the scenario where a catcher has not been detected by the currently active set of rules.

After the evaluation has been completed, the picture on the bottom will change to reflect the result found. Additionally, if PCH scan integration is enabled, the results from *User Scan* will also carry over to the data model if a PCH scan has been carried out in the process.

## 3.4. Related Projects

IMSI catcher detection is a topic that has not emerged until recently, therefore, not a lot of work and research has been done upon that subject. This is mainly due to the fact that it was hard to get information from the mobile network onto a computer for evaluation and the threat seemed to be not as large as today with cheap self build IMSI catchers available.

About the same time as this project, in December 2011, another project was announced with the same goals of detecting an IMSI catcher. The project is called 'Catcher Catcher'[1] and also builds upon the OsmocomBB framework. The goals are the same, however, the means are very different. As a codebase 'Catcher Catcher' uses the `mobile` application, a software that implements the firmware part of a mobile phone. This results in an active approach to IMSI catcher detection. An active connection is established between the phone and the base station in question. Basically this means that identification is done by letting a bait-phone get caught.

The advantage compared to the passive approach this project uses is that one has more sure means at hand of identifying a potential catcher. Features that are already implemented are [18]:

- Encryption: Check whether encryption is enabled when doing a phone call.
- IMEI: IMEI is not requested in Cipher Mode Complete message.
- LAC: LAC of a base station changes.

[1] Catcher Catcher Wiki, `http://opensource.srlabs.de/projects/catcher/wiki/Tutorial` [Online; Accessed 05.2012]

- Location Updates: IMEI is requested during Location Updates.
- Silent Text: Checks whether a silent text message is received.
- Call Setup: Do not receive a Call Setup message while being on a traffic channel for two seconds.

As one can see, missing encryption and reception of a silent text message are very strong indicators of being connected to a catcher. This however comes at the cost of being discovered oneself.

It is not clear whether the project has been abandoned or whether it is developed further. Activity on the Wiki and Git has seized after December 2012.

# 4. Evaluation

The following chapter presents the results of the experiments, carried out with the ICDS. Evaluation has been done in different areas to give a complete impression of how the ICDS performs. In the first section, some general findings will be described that affect overall performance. Afterwards, the test environment and setup of the IMSI catcher is discussed. The last section describes the evaluation of the ICDS against a configured catcher, performing different attacks.

## 4.1. Performance Evaluation

In order to evaluate general performance, it has to be considered that the ICDS can be deployed in different environments. To reflect different compositions and densities of base stations from different areas, four distinct data sets will be used for the experiments in this section. The data sets have been taken in areas surrounding the city of Freiburg. For each area, three scans were made on a fixed position and the duration was averaged. Table 4.1 shows some of the data sets' key values.

Apart from nodes of the four German GSM providers E-Plus, T-Mobile, Vodafone and $O_2$, nodes from the Deutsche Bahn also occur in these scans. These nodes form a private network, used for internal communications by the Deutsche Bahn. They are identified by their broadcast name *DB Systel GSM-R* and their frequency which is a in a range registered to the Deutsche Bahn. Since the distribution of these nodes is very sparse, only

| Name | Description | Number of BTS | Duration |
|---|---|---|---|
| `cdb` | CBD around the area of Bertoldsbrunnen | 54 | 6:13 m |
| `airport` | Airport and university area around Georges Köhler Allee | 68 | 6:25 m |
| `ind_park` | Industrial park Haid in Freiburg West, Hausener Weg | 53 | 4:52 m |
| `house_area` | Housing area at the rim of Freiburg Zähringen, Thuner Weg | 22 | 3:59 m |

Table 4.1.: Key values of the data sets used for performance tests.

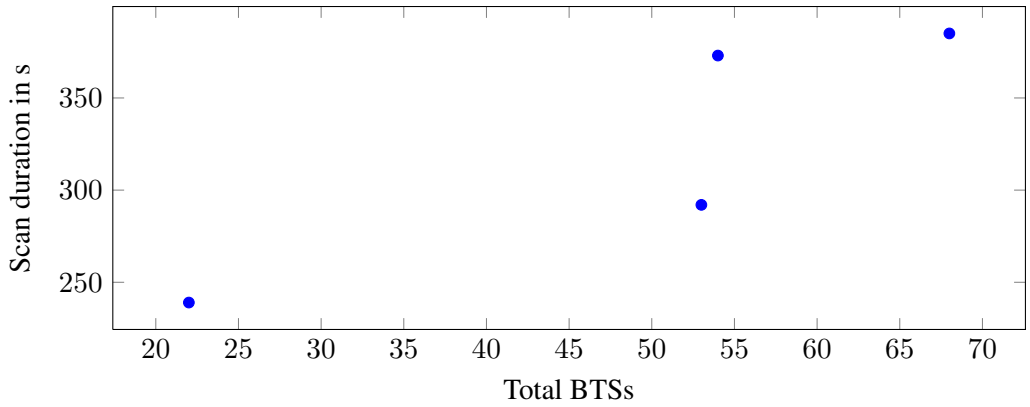



Figure 4.1.: Scan durations for the sample data sets. From left to right the datasets are: `house_area`, `ind_park`, `cbd`, `airport`

one node can be found in each scan. They yield a false positive for no neighbouring nodes can be discovered. These nodes are not relevant to subscribers because they are not able to connect to them. Therefore, they will be ignored and factored out for the remainder of this evaluation.

### 4.1.1. Scan Duration

Table 4.1 shows that the time needed for a sweep scan in the Freiburg area can differ by large amounts, depending on how many base stations have been scanned. Generally said, it takes longer, the more dense the base station distribution is in the area. This is however not the only factor, as Figure 4.1 visualises. If the scan duration would only depend on the number of base stations scanned, a linear growth could be expected.

This is however not the case as the plot shows. A bad reception means that a lot of BCCH frames are rendered unusable and have to be retransmitted. It takes significantly longer to gather all System Information Messages for a single BTS that has a bad reception. Looking at the overall reception in the datasets shows that no base stations in the `house_area` dataset had a reception of below -95 dB. In the three other datasets, stations with reception levels of below -100 dB can be found. Overall reception was worst in the `airport` and `cbd` datasets which explains the large jump in time although only one more base station has been scanned between the `ind_park` and `cbd` datasets.

Re-evaluation of a base station, based on its own parameters thus occurs only every seven minutes in the worst scenario we experienced. This is an inherent problem to the approach of scanning and updating all base stations and not only monitoring a subset belonging to a single provider. If an IMSI catcher replaces a base station directly after it was scanned, it could take up to seven minutes until it is discovered. To lessen this threat,

| | cdb | | airport | | ind_park | | house_area | |
|---|---|---|---|---|---|---|---|---|
| | Cov. | Time | Cov. | Time | Cov. | Time | Cov. | Time |
| Google | 1.00 | 5 | 0.99 | 8 | 1.00 | 5 | 1.00 | 2 |
| OCID | 0.57 | 51 | 0.58 | 68 | 0.58 | 55 | 0.41 | 19 |

Table 4.2.: Coverage for Google Mobile Maps and OpenCellID on the data sets with the time needed in seconds for fetching the information.

if the ICDS is used in *User Mode*, the base station with the strongest reception is scanned again with a PCH scan, to eliminate the possibility of having been taken over and not being detected.

### 4.1.2. Cell ID Databases

The usefulness of the Cell ID Rule is subject to the completeness of the database that is used. That is even more so since a database with a low coverage will yield false positives, e.g. legitimate base stations will be evaluated as being IMSI catchers because they are not found in the database.

The coverage for the OpenCellID database and the Google Mobile Maps service evaluated against the data sets can be seen in Table 4.2. Google Mobile Maps service scored a complete coverage on all the data sets while OpenCellID could cover about half the nodes in the different sets. The Ericson and combain databases could not be evaluated since it was not possible to obtain an API key without handing out credit card details for billing. The reason the Google service only had a 99% coverage on the `airport` data set is that the base station that has not been found was the one operated by the Chair of Communication Systems, therefore, this is not a problem. The OpenCellID database is not a good source of information for this project as is shown by its coverage scores. Both services also show a large difference in response time. The time needed to do a single lookup on OpenCellID could take up to several seconds while a single lookup on the Google service presented a result almost instantly. This is most probably due to the fact that Google's server infrastructure is strongly optimised for tasks like this. The times also show that if the ICDS would be connected to the internet, the lookups on Google's database could also be done during the course of a sweep scan since they do not impose a large time overhead per base station.

However, it must be said that these two services are intended for localisation and thus do not have the demand to yield a complete coverage of all the base stations in the area. It must be kept in mind when using this rule for analysis that false positives might still be brought forth when using online services. What can be said though is that a base station that has been found, may only be subject to a type of attack that replaces an existing base

station and can thus be investigated more specifically on that ground.

### 4.1.3. PCH Scans

In order to establish a baseline on what to expect from the PCH scans, additional measurements have been done. Table 4.3 shows scans that have been done in the different areas. In each area, the cell with the strongest reception for each provider was chosen as a representative for the respective provider. The duration of each scan was set to 60 s, while the values in the table have been averaged for 10 s for this is the unit the ICDS is using.

A comparison of the results suggests that different providers also have different policies when to page. Vodafone has about six times the paging rate of other providers. This can be explained by further examining the Vodafone network structure. Another scan showed that for other providers, the Paging Messages were addressed to between 70 and 120 different TMSIs whereas for Vodafone between 600 and 700 different TMSIs were found. The large difference in TMSIs is due to the fact that Vodafone's LAs are larger than the LAs other providers use. For the Freiburg area two different LACs were found for each of the providers E-Plus, T-Mobile and $O_2$ while for Vodafone only one LAC was found. These facts were also checked against the OpenCellID database which yielded the same results for LACs used in the Freiburg area. All this gives some insights into the paging policy that Vodafone might have. If the network is looking for a subscriber the last known LA for this subscriber is paged rather than starting with the last known cell and expanding the paging radius. Since the area covered by a single LA is very large, a lot of subscribers are registered for a single area. This theory would also be consistent with the fact that despite of the large number of Paging Messages, only an average number of IAs were caught which are restricted to the serving cell.

Another scan was also done on the IMSI catcher. No Paging Messages or IAs were detected although a MS was connected to it. This was to be expected as formerly discussed in Section 3.2.2 because the IMSI catcher is not actually part of the providers network and thus cannot receive and forward Paging Messages.

## 4.2. IMSI Catcher Detection

Before using an IMSI catcher for testing purpose or a launching an OpenBTS base station, it should be ensured that licenses for the specific frequencies that are used, have been obtained. This way, the operation of these devices does not interfere with regular radio communication. In case of our experiments we always used ARFCN 877, for which the university has acquired a license. The identification we broadcasted was '23' to not accidantly lure subscribers into trying to connect.

| | house_area | | cbd | | airport | | ind_area | |
|---|---|---|---|---|---|---|---|---|
| | PMs. | IAs | PMs. | IAs. | PMs. | IAs. | PMs. | IAs |
| T-Mobile | 89 | 3 | 75 | 3 | 109 | 4 | 72 | 1 |
| E-Plus | 119 | 1 | 67 | 2 | 70 | 1 | 65 | 0 |
| Vodafone | 776 | 6 | 720 | 5 | 712 | 6 | 743 | 2 |
| $O_2$ | 117 | 9 | 106 | 16 | 94 | 11 | 95 | 7 |

Table 4.3.: Number of Paging Messages and Immediate Assignments (per 10 s) for the four German providers at different locations.

### 4.2.1. Open Source IMSI Catcher

Some of the rules cannot be tested without an active IMSI catcher. For this purpose the Open Source IMSI-Catcher [27] is used.

This project prototypes an IMSI catcher using only open source systems and freely available hardware, so it can be used and built by anybody. On the hardware side a computer running a Linux operating system is used, as well as the Universal Software Radio Peripheral (USRP) as the radio transmitter. The USRP allows the signal processing for radio transmissions to be done in software, therefore, it can be used for a multitude of purposes and protocols. Some hardware modifications have to be done to the device to empower it to send and receive data on the frequency bands used for GSM communication. An external clock needs to be used since GSM operations are very time critical. Figure 4.2 shows the Open Source IMSI Catcher and the ICDS side by side.

On the software side GNU Radio[1], OpenBTS[2] and Asterisk[3] are used to achieve the functionality provided by an IMSI catcher. The raw data that is received by the USRP is sent to the GNU Radio component which works as a software side interface to the USRP. This data is taken by the OpenBTS software that simulates base station behaviour and has an integrated module simulating a VLR handing out TMSIs. OpenBTS implements an open source version of the GSM stack with the goal to provide cheap access points to the GSM network in areas with bad coverage. The user accounts as well as encoding of voice data and recording of calls is handled inside the Asterisk software, basically combining the TRAU, HLR and authentication centre of a real GSM network. Calls are routed from here on to the Voice over IP (VoIP) network of the university.

Since we do not want to actually connect to the IMSI catcher, the Asterisk part and user configuration will be omitted here. The parameters necessary to simulate a GSM cell have to be set inside the `OpenBTS.conf`. Figure 4.3 shows an annotated example for a

[1]GNU Radio Project Wiki, `http://gnuradio.org/redmine/projects/gnuradio/wiki` [Online; Accessed 05.2012]

[2]OpenBTS Project Wiki, `http://wush.net/trac/rangepublic` [Online; Accessed 05.2012]

[3]Asterisk, `http://www.asterisk.org` [Online; Accessed 05.2012]

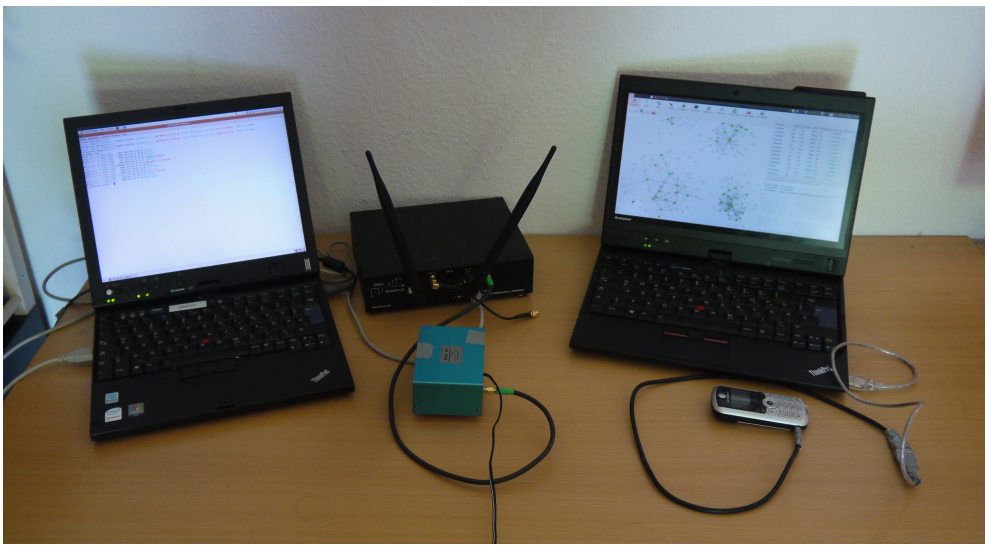


Figure 4.2.: Open Source IMSI Catcher (left) with USRP (black) and external clock (blue) and the ICDS (right) with the Motorola C123 connected.

configuration which would simulate a T-Mobile cell. `Control.OpenRegistration` is explicitly set to 0 which prevents anyone from connecting to the IMSI catcher since connections are not part of the test and we do not want to interfere with other peoples' communications in the area. More precisely, this will only let users connect that have been set up in the `sip.conf` of the Asterisk server. Only the test phone does have a valid account.

As a general note, when the experiments were conducted the ICDS and the Open Source IMSI-Catcher were located in the same room, therefore the IMSI catcher had always good reception levels. This is not a problem since an IMSI catcher operator generally wants to have high reception levels on the target phone to lure it to connect to the device. So if the IMSI catcher would be located farther away the operator would increase transmission power accordingly.

### Modifications to the ICDS Configuration

A few small modifications have to be made to the configuration of the ICDS to not instantly evaluate the university base station and the IMSI catcher as *Critical*.

'23' is the provider name broadcasted by the university base station. The configuration of the ARFCN / Provider Mapping Rule has been changed to include the ARFCN 877 as valid ARFCN for the imaginary provider '23'. The country 'Germany' was also added to the dictionary as a valid country for provider '23'. Furthermore '23' was included in the

```
#Do not let people connect
Control.OpenRegistration 0

#Basic cell parameters
GSM.MCC 262                         GSM.ARFCN 54
GSM.MNC 01                          GSM.ShortName T-Mobile
GSM.LAC 29184
GSM.CI 61858

#Transmission strength ranging from 0 to 23
GSM.PowerAttenDB 20

#Neighbouring cell list, space separated
GSM.Neighbours 69 53 20

#Force location Updates, multiple of 6 minutes
GSM.T3212 1
```

Figure 4.3.: Excerpt of a `OpenBTS.conf`.

list of known providers which the Provider Known Rule uses and 4711 was included as a valid LAC for this provider.

Another small change has been done to the implementation of the Neighbourhood Structure Rule to treat the provider '23' as an equivalent to E-Plus. This has been done because the university base station has E-Plus nodes as neighbours which would normally trigger a *Critical* rating on the Neighbourhood Structure Rule. On the other hand this needed to be done so we could add E-Plus neighbours to the catcher cell in order to have a valid neighbouhood list when needed.

### 4.2.2. Configuration and Context Rules Evaluation

With the environment set up we will now evaluate the individual Rules. The IMSI catcher was launched with the three different configurations 2–4 shown in Table 4.4.

Configuration 1 will now be used to recap the rules theoretically since we cannot actually transmit on ARFCN 50. Table 4.5 summarises and explains the findings of the different Configuration and Context Rules for this imaginary scenario. The Neighbourhood Structure Rule should be given a closer examination. Since neighbours are present and at least one neighbour has been found directly, the basic requirements for the rule to yield an *Ok* have been met. However since its ARFCN is 50, it has no incoming edges in

| | Conf. 1 | Conf. 2 | Conf. 3 | Conf. 4 |
|---|---|---|---|---|
| ARFCN | 50 | 877 | 877 | 877 |
| ShortName | T-Mobile | 23 | 23 | 23 |
| MCC | 262 | 262 | 262 | 505 |
| MNC | 01 | 23 | 23 | 23 |
| LAC | 21010 | 123 | 4711 | 4711 |
| Cell ID | 1 | 2 | 3 | 19279 |
| Neighbours | 42, 44, 45 | 778, 779, 780 | 818, 695, 828 | 977, 997, 992 |

Table 4.4.: Erroneous configurations for the IMSI catcher.

| Rule | Finding | Explanation |
|---|---|---|
| Provider Known | *Ok* | T-Mobiled is a known provider. |
| Country / Provider Map | *Ok* | MNC 262 and MNC 01 with T-Mobile fit together. |
| LAC / Provider Map | *Critical* | LAC 21010 not a known LAC for MNC 01 in the Freiburg area. |
| ARFCN / Provider Map | *Critical* | ARFCN 50 belongs to Vodafone. |
| LAC Median Deviation | *Critical* | LAC differs from other T-Mobile stations in the area. |
| Pure Neighbourhoods | *Ok* | Only T-Mobile stations as neighbours. |
| Neighbourhood Structure | *Warning* | Explanation in running text. |
| Discovered Neighbours | *Ok* | All neighbours have been discovered. |
| Cell ID Uniqueness | *Ok* | No duplicate Cell ID found. |

Table 4.5.: Configuration and Context Rule results for Config 1.

the neighbourhood graph from other T-Mobile nodes thus the rule only yields a *Warning* result.

With each of the remaining configurations, the ICDS detected the catcher for various reasons. All rules mentioned did yield a *Critical* rating unless noted otherwise.

- Config 2: The detected errors within this configuration are that none of the neighbours mentioned was in range to be detected, which is very unlikely for a normal base station. Additionally LAC 123 is not a known LAC for '23'. Rules triggered: Neighbourhood Structure, LAC / Provider Map, Discovered Neighbours Rule.

- Config 3: In this configuration one of the neighbours, namely 695 ($O_2$) is not consistent with the set provider ('23' / E-Plus). The base station breaks up the isolated subgraph structure for E-Plus and is thus detected.
  Rules triggered: Pure Neighbourhoods

- Config 4: The chosen provider is not consistent with the country set. Additionally another warning is thrown since the neighbourhood list only contained nodes that were found indirectly.
  On top of that, the CID was already in use by another station. Rules triggered: Country / Provider Mapping, Neighbourhood Structure (*Warning*), Cell ID Uniqueness.

### 4.2.3. Scan Rules Evaluation

For the purpose of testing the LAC Change and rx Change Rules the procedure was as follows. At first the ICDS was turned on and scanning commenced. Afterwards the IMSI catcher was turned on, operating on ARFCN 877, the same as a base station that was previously discovered. This was repeated several times with different configurations of the IMSI catcher. Table 4.6 summarises the findings. The configurations used can be found in Appendix E.1. In all cases the ICDS was able to detect the IMSI catcher after about 6 minutes which corresponds to the time that is needed to conduct a complete sweep scan. These times can vary however depending on the timing of the catcher being turned on and the time it takes for rescanning a base stations as described in the beginning of this chapter.

### 4.2.4. Database Rules Evaluation

To evaluate the Local Area Database Rule and Cell ID Database Rule a long-term test has been carried out. This has been done to find out whether base stations in the surrounding area change on a regular basis or stay the same (including their respective configurations and reception levels). This is essential for databases to be usable over a longer period of time.

| | rx | | LAC | | | | |
|---|---|---|---|---|---|---|---|
| Config | Old | New | Old | New | rx det. | LAC det. | Time |
| Config 5 | -92 dB | -44dB | 4711 | 666 | Yes | Yes | 6:31 m |
| Config 6 | -91 dB | -46dB | 4711 | 4711 | Yes | No | 6:22 m |
| Config 5 | -89 dB | -44dB | 4711 | 666 | Yes | Yes | 5:59 m |
| Config 6 | -92 dB | -43dB | 4711 | 4711 | Yes | No | 6:35 m |

Table 4.6.: Results obtained testing the *rx* and *LAC Change rules*.

The database itself has been built over the course of one week in Freiburg, Georges Köhler Allee. Two scans were conducted per day and integrated with the ICDS into the existing Local Area Database. During this period no parameter changes were detected and the reception of base stations only varied inside a very small interval.

After that, each day for another one and a half weeks, two scans per day were done, one at around 11:00 am and one around 8:00 pm. One of them was conducted while the IMSI catcher was operating, the other without the device present. The gap between the 5$^{th}$ and the 8$^{th}$ was due to the face the IMSI catcher was unavailable during these days. The results on a per day basis are summarised in Table 4.7.

Two different configurations for the IMSI catcher were in place each, targeting one of the rules. In cases it was detected by the Local Area Database Rule a configuration was used that mirrored the base station that was replaced. In the other cases where the Cell ID Database Rule triggered the same configuration was used, but the CID was changed to be a new one. The catcher and the normal base station were sending at the same frequency for these cases since the base station could not be switched off. No problems occurred due to that fact since the IMSI catcher had a significantly better reception and was found in all cases instead of the regular base station.

During this two and a half week time period in which the databases were built and the tests done, none of the BTSs in the surrounding area listed a significant change in reception or parameters. Therefore no false positives or negatives had been found. All cases in which the IMSI catcher was operating were found either because the reception on the frequency was exceptionally good or because the CID used was not in the database.

### 4.2.5. Realistic Scenarios

Since all the rules have been tested we assume from this point on that the IMSI catcher is configured correctly, meaning that parameters like the ARFCN, LAC and provider have been set up in correct and consistent way so the respective rules will not show an alarm. Consistent parameters for the four providers in Germany can be found in Table 4.8. Note that the CID can be a arbitrary value as long as it is unique in the area of reception. CIDs

| Date | Time | Catcher | Detected | Detected by |
|---|---|---|---|---|
| 31.05.12 | 11:00 am | Yes | Yes | Local Area Database |
| 31.05.12 | 8:00 pm | No | No | |
| 01.06.12 | 11:00 am | No | No | |
| 01.06.12 | 8:00 pm | Yes | Yes | Cell ID Database |
| 02.06.12 | 11:00 am | Yes | Yes | Local Area Database |
| 02.06.12 | 8:00 pm | No | No | |
| 03.06.12 | 11:00 am | No | No | |
| 03.06.12 | 8:00 pm | Yes | Yes | Cell ID Database |
| 04.06.12 | 11:00 am | Yes | Yes | Local Area Database |
| 04.06.12 | 8:00 pm | No | No | |
| 05.06.12 | 11:00 am | No | No | |
| 05.06.12 | 8:00 pm | Yes | Yes | Cell ID Database |
| 08.06.12 | 11:00 am | Yes | Yes | Local Area Database |
| 08.06.12 | 8:00 pm | No | No | |
| 09.06.12 | 11:00 am | No | No | |
| 09.06.12 | 8:00 pm | Yes | Yes | Cell ID Database |

Table 4.7.: Results of the database evaluation.

| Parameter | T-Mobile | Vodafone | E-Plus | $O_2$ |
|---|---|---|---|---|
| ARFCN | 13-49, 81-102, 122-124, 587-611 | 1-12, 50-80, 103-121, 725-751 | 975-999, 777-863 | 0, 1000-1023, 637-723 |
| LAC | 21014/21015 | 793 | 588/138 | 50945/51903 |
| MCC | 262 | 262 | 262 | 262 |
| MNC | 01 | 02 | 03 | 07 |

Table 4.8.: Consistent parameter configurations in the Freiburg area for the four German providers.

measured from different base stations do not follow any particular schema. The scenarios are built after the attacks described in Section 2.4.1. Local information in terms of a Local Area Database was available.

### IMSI Catcher as a new Cell

The first scenario will simulate the case where the catcher opened up a new cell with a good reception and forced the MS into normal cell selection mode by disconnecting it from the current base station via a jammer. First the IMSI catcher was turned on, faking a legitimate '23' cell with a new CID. Afterwards the ICDS was started and a sweep scan was performed. As soon as the cell was scanned which occurred very early since the reception was very good (-45 dB) it was detected that this cell was not in the Local Area Database. After the sweep scan CIDs, from Google were also fetched. Both the Local Area Database Rule and the Cell ID Database Rule indicated a *Critical* status.

As a further step to simulate the case where no local information is available, the Local Area Database Rule and Cell ID Rules were turned off. The ICDS then yielded an *Ok* evaluation since the configuration of the catcher cell was consistent. The next step was to put the ICDS into *User Mode* with '23' as its fixed provider. It selected the IMSI catcher cell as its target cell because of the good reception level and since its evaluation was *Ok*, an additional PCH scan was started. No paging messages or IAs were caught so the end result was a *Critical* status for the IMSI catcher cell.

### IMSI Catcher replacing an old Cell

The second scenario simulated the attack where the IMSI catcher replaces a base station with a bad reception, in the neighbourhood of the cell the MS is connected to. This way the reception drastically improves on that particular frequency suggesting to the MS that the subscriber moved into the close perimeter of that BTS and it switches its cell to the stronger one.

We used the university base station on ARFCN 877 as our target. A sweep scan was conducted with the ICDS and after the base station had been found the IMSI catcher was started on the same frequency.

Due to its strong increase in reception and the change in the LAC, the IMSI catcher cell obtained a *Critical* status immediately after ARFCN 877 had been scanned a second time by the two Scan Rules. Also due to the fact the reception level differed too much from the interval that had been measured for this CID in the Local Area Database Rule also yielded a *Critical* rating. *User Mode* did not start a PCH scan since the evaluation had already been *Critical*.

# 5. Conclusion

This chapter will give a short summary of the whole project and its findings. The first section starts by reviewing what has been done while the second section will then bring up some aspects where the ICDS could be improved to yield results either faster or more accurate.

## 5.1. Summary

The aim of this project was to find ways of unveiling whether an IMSI catcher is being operated in the close perimeter or not. In other words whether it is safe to connect to the GSM network and place phone calls or not. An unsafe environment could result in IMSI numbers being requested and saved by IMSI catchers or in phone calls being recorded. The main premise that distinguishes this project from other similar projects like the also OsmocomBB based 'Catcher Catcher', is that the system is operating in a completely passive manner. Therefore it can only work on a limited amount of information, namely on information that is broadcasted on publicly available channels. The benefit this yields over other projects is that the IMSI Catcher Detection System itself is completely invisible to the IMSI catcher.

Chapter 2 laid out basic concepts of GSM communication to create a basis for understanding why and how an IMSI catcher works. Some more detailed concepts on the $U_m$ interface were discussed to enable the reader to grasp the concept of logical channels and how they can be used to harvest information in a passive manner. The chapter concluded with an account of how an IMSI catcher operates, by outlining the two main ways of attacking a subscriber — one by creating a new cell for the subscriber to connect to and the other by overtaking an already existent cell.

Chapter 3 started by explaining how the OsmocomBB framework was used to build the ICDS. It concluded with a summary of how to configure and use the system. The two main sources of information, the BCCH and the PCH were introduced along with the different parameters that the ICDS bases its findings on. An outline of how a finding is reached is illustrated in Figure 5.1. At first a sweep scan is conducted or an old project is loaded to supply the ICDS with information about surrounding base stations. During the scan or after the data has been loaded, the ICDS evaluates different rules on the data. This can be done with or without consulting databases containing local information. The results show that some IMSI catcher configurations can be uncovered by these rules

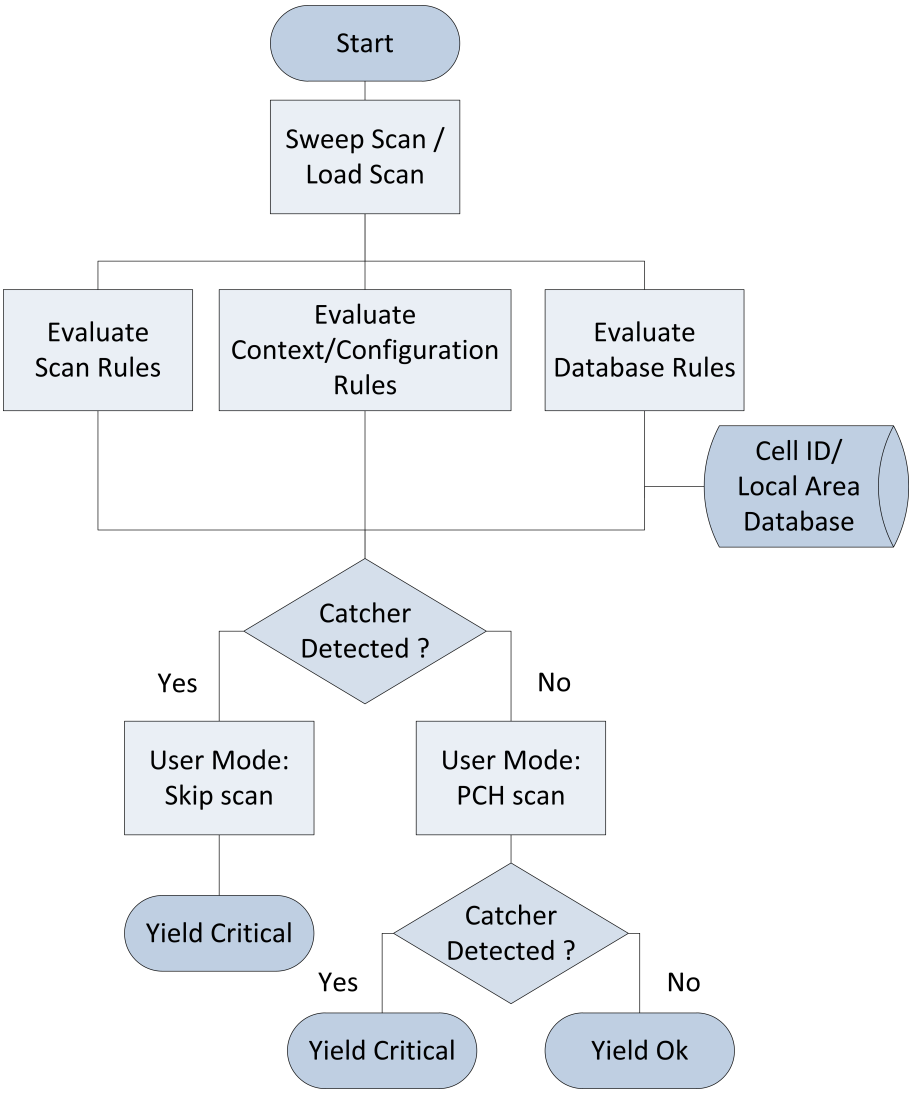



Figure 5.1.: ICDS decision finding process outlined.

which check basic configuration data obtained from System Information Messages. In addition to this data broadcasted on the BCCH, reception levels and LACs are also monitored over time to unveil attacks in which existing base stations are replaced by IMSI catchers. This leaves IMSI catchers that have a consistent configuration and blend well in their surroundings concerning the reception levels. They are also broadcasting the same LAC as the replaced base station, even if this means it could take a long time until the MS announces itself. To handle this case, the ICDS can monitor the PCH of the base station in question to gather Paging Messages and Immediate Assignments. Since an IMSI catcher is not part of the provider's network, no Paging Messages will be forwarded to the connected subscribers. These findings have been confirmed with the experiments performed in Chapter 4 where different attack scenarios have been tested. In cases where the ICDS was not able to uncover the IMSI catcher by rule evaluation, the PCH scan yielded the desired result. It should be kept in mind that the evaluation has been done against a prototype IMSI catcher since data from a real IMSI catcher is not available. However, the results provided in this thesis are based more on general procedures, the GSM protocol itself and not tailored to the specific system. Therefore they should be applicable to any IMSI catcher that uses the attacks outlined here.

## 5.2. Future Work

There are several ways in which the ICDS could be improved. The experiments showed that one of the main issues is the duration of the sweep scans. If a BTS is replaced right after it has been scanned, it can take up to seven minutes until it is scanned again and the IMSI catcher is uncovered. That is the time that is needed to do a complete sweep scan. The ICDS could be refined so that only base stations of a particular provider are monitored This would cut down the time to conduct sweep scans significantly, it could also be done upon entering *User Mode*.

In case of the Open Source IMSI-Catcher no Paging Messages were sent. However, it would be possible for a catcher that is aware of this evaluation criterion, to send fake Paging Messages to arbitrary TMSIs to deceive the ICDS. To face this, the ICDS could be extended. Since Paging Messages would be unreliable in such a case, one would have to use IAs. The experiments have shown that this might increase scanning time on the PCH since these messages are much more rare than Paging Messages. An IA sent to a subscriber contains the dedicated channel on which the conversation between the base station and the mobile phone is to continue. At this point, the ICDS already uses the information about dedicated channels to see whether frequency hopping is used or not. If an IA is caught by the ICDS one could follow on the assigned channel and catch the Cipher Mode Message. An IMSI catcher will disable encryption to tap into calls, the Cipher Mode Message would contain A5/0 as its encryption algorithm instead of A5/1 which is used in Germany. This feature could be used to handle cases of fake

Paging Messages or IAs, however, it would take longer to conduct the PCH scan. Another problem would be that it requires another subscriber that is connected to the IMSI catcher initiating a call. On the other hand a regular base station using encryption can also be verified this way.

This thesis analysed the threats that an IMSI catcher poses and which security flaws are responsible for its success. The main security flaw used in GSM, namely authentication not being mutual, is fixed in the next generation UMTS networks. The topic will, nevertheless, be of importance for years to come since as long as UMTS coverage is not complete, mobile phones will continue to have a fallback mechanism to look for GSM cells when no UMTS cells are available. To force a mobile phone to fall back to GSM an IMSI catcher operator could jam the UMTS frequency band and wait until the MS connects.

The results show that it is possible to identify suspicious base stations and in this way lessen the thread of being caught.

We presented, with the ICDS, a step into the direction of more security aware systems. A step that could and should also be taken by telephone manufacturers when designing the firmware and operating system for their next device.

# Bibliography


[1] 3GPP TECHNICAL SPECIFICATION GROUP CORE NETWORK AND TERMINALS. Numbering, addressing and identification. *TS 23.003*, *DOC file*, http://www.3gpp.org/ftp/Specs/archive/23_series/23.003/23003-a30.zip, September 2011.

[2] 3GPP TECHNICAL SPECIFICATION GROUP GSM/EDGE RADIO ACCESS NETWORK. Data link (DL) Layer: General aspects. *TS 04.05*, *DOC file* http://www.3gpp.org/ftp/Specs/archive/04_series/04.05/0405-802.zip, May 2002.

[3] 3GPP TECHNICAL SPECIFICATION GROUP GSM/EDGE RADIO ACCESS NETWORK. Multiplexing and Multiple Access on the Radio Path. *TS 05.02*, *DOC file* http://www.3gpp.org/ftp/Specs/archive/05_series/05.02/0502-8b0.zip, June 2003.

[4] 3GPP TECHNICAL SPECIFICATION GROUP GSM/EDGE RADIO ACCESS NETWORK. Radio Access Network: Radio transmission and reception. *TS 05.05*, *DOC file*, http://www.3gpp.org/ftp/Specs/archive/05_series/05.05/0505-8k0.zip, November 2005.

[5] 3GPP TECHNICAL SPECIFICATION GROUP GSM/EDGE RADIO ACCESS NETWORK. Mobile Station - Base Station System (MS - BSS) interface: Data Link (DL) layer specification. *TS 04.06*, *DOC file*, http://www.3gpp.org/ftp/Specs/archive/04_series/04.06/0406-840.zip, December 2008.

[6] 3GPP TECHNICAL SPECIFICATION GROUP GSM/EDGE RADIO ACCESS NETWORK. Mobile radio interface layer 3 specification: Radio Resource Control (RRC) protocol. *TS 44.018*, *DOC file*, http://www.3gpp.org/ftp/Specs/html-info/25321.htm, March 2012.

[7] 3GPP TECHNICAL SPECIFICATION GROUP RADIO ACCESS NETWORK. Medium Access Control (MAC) protocol specification. *TS 25.321*, *DOC file*, http://www.3gpp.org/ftp/Specs/html-info/25321.htm, December 2011.

[8] 3GPP TECHNICAL SPECIFICATION GROUP RADIO ACCESS NETWORK. UE Radio Access capabilities. *TS 25.306*, *DOC file*, http://www.3gpp.org/ftp/Specs/html-info/25306.htm, March 2012.

[9] CHAUDHURY, P., MOHR, W., AND ONOE, S. The 3GPP proposal for IMT-2000. *Communications Magazine, IEEE 37*, 12 (1999), 72–81.

[10] EBERSPÄCHER, J., VÖGEL, H.-J., BETTSTETTER, C., AND HARTMANN, C. *GSM – Architecture, Protocols and Services*. Wiley, 2009.

[11] ETSI. Digital cellular telecommunications system (Phase 2+): Mobile Stations (MS) features. *TS 02.07*, *DOC file*, `http://www.3gpp.org/ftp/Specs/archive/02_series/02.07/0207-710.zip`, March 2000.

[12] FEDERRATH, H. Protection in mobile communications. In *Multilateral Security in Communications – Technology, Infrastructure, Economy* (1999), G. Müller and K. Rannenberg, Eds., Addison-Wesley-Longman, pp. 349–364.

[13] FOX, D. Der IMSI-catcher. *Datenschutz und Datensicherheit 26*, 4 (2002), 212–215.

[14] GLOBAL MOBILE SUPPLIERS ASSOCIATION. GSM/3g Stats. *WWW document*, `http://www.gsacom.com/news/statistics.php4`, [Online; Accessed 06.2012].

[15] GSM ASSOCIATION. Brief History of GSM and the GSMA. *WWW document*, `http://www.gsma.com/aboutus/history/`, [Online; Accessed 06.2012].

[16] HAUG, T. *Overview of GSM: philosophy and results*. *International Journal of Wireless Information Networks 1*, 1 (1994), 7–16.

[17] HEINE, G. *GSM networks: Protocols, Terminology, and Implementation*. Artech House, 1999.

[18] OSMOCOMBB. Catcher Catcher. *Project Wiki*, `http://opensource.srlabs.de/projects/catcher/wiki`, [Online; Accessed 01.2012].

[19] OSMOCOMBB. Motorola C123. *Project Wiki*, `http://bb.osmocom.org/trac/wiki/MotorolaC123`, [Online; Accessed 06.2012].

[20] OSMOCOMBB. Project Rationale. *Project Wiki*, `http://bb.osmocom.org/trac/wiki/ProjectRationale`, [Online; Accessed 06.2012].

[21] RIES, U. IMSI-Catcher für 1500 Euro im Eigenbau. *WWW document*, `http://heise.de/-1048919`, August 2010.

[22] SAFFERLING, C. Terror and law. *Journal of International Criminal Justice 4*, 5 (2006), 1152–1165.

[23] SAUTER, M. *Grundkurs mobile Kommunikationssysteme : von UMTS, GSM und GRPS zu Wireless LAN und Bluetooth Piconetzen*. Vieweg, 2006.

[24] SCOURIAS, J. Overview of GSM: The global system for mobile communications. *University of Waterloo*, *PDF file*, `http://ccnga.uwaterloo.ca/publications/pdfs/TR-96-01.pdf`, 1996.

[25] TELECOMUNICATION STANDARDIZATION SECTOR OF ITU. General series Intelligent Network Recommendation structure. *Recommendation Q1200*, *DOC file*, `http://www.itu.int/rec/T-REC-Q.1200-199709-I/en`, September 1997.

[26] TELECOMUNICATION STANDARDIZATION SECTOR OF ITU. List of Mobile Country or Geographical Area Codes. *Complements to Recommendation E.212*, *PDF file*, `http://www.itu.int/itudoc/itu-t/ob-lists/icc/e212_685.html`, January 2004.

[27] WEHRLE, D. Open Source IMSI-Catcher. *Master Thesis at the Chair of Communication Systems at Freiburg University*, October 2009.

[28] WELTE, H., AND MARKGRAF, S. OsmocomBB - Running your own GSM stack on a phone. *PDF file*, `http://events.ccc.de/congress/2010/Fahrplan/attachments/1771_osmocombb-27c3.pdf`, July 2010.

[29] WIKIPEDIA. Cell ID. *WWW document*, `http://en.wikipedia.org/wiki/Cell_ID`, [Online; Accessed 02.2012].

[30] WIKIPEDIA. IMSI-Catcher. *WWW document*, `http://de.wikipedia.org/wiki/IMSI-Catcher`, [Online; Accessed 02.2012].

[31] WIKIPEDIA. Equipment Identity Register. *WWW document*, `http://en.wikipedia.org/wiki/Central_Equipment_Identity_Register`, [Online; Accessed 06.2012].

[32] ZAHORANSKY, R. Localization in GSM Mobile Radio Networks . *Master Thesis at the Chair of Communication Systems at Freiburg University*, November 2011.

# List of Figures

# List of Tables

# Appendix A.

# GSM

## A.1. Interfaces

The following table contains a brief description of the interfaces used inside a GSM network. On the upper part the interfaces for the Network Subsystem are listed and on the lower part the interfaces for the Base Station Subsystem can be found.

| Name | Between | Function |
|---|---|---|
| $A$ | MSC $\leftrightarrow$ BSS | BSS management data for Mobility Management and Call Control |
| $B$ | MSC $\leftrightarrow$ VLR | MSC receives data about MSs in the current area and sends data from Location Updates |
| $C$ | MSC $\leftrightarrow$ HLR | MSC can request routing data during call setup and send e.g. charging information |
| $D$ | HLR $\leftrightarrow$ VLR | Exchange of location-dependent subscriber data and updating the HLR (MSRN etc. ) |
| $E$ | MSC $\leftrightarrow$ MSC | Executing a Handover when subscriber changes to a new MSC |
| $F$ | MSC $\leftrightarrow$ EIR | Checking white-/grey- and blacklists before giving access to the network |
| $G$ | VLR $\leftrightarrow$ VLR | Connects VLR of different MSCs to exchange subscriber data during a handover |
| $A_{\text{bis}}$ | BSC $\leftrightarrow$ BTS | BSC receives data from MS via the BTS |
| $U_m$ | BTS $\leftrightarrow$ MS | Registration procedure, call data etc. as well as broadcast information about the network and the base station |

Table A.1.: Interface found in the GSM network.

## A.2. Channel Combinations

The following table contains the possible combinations of channels inside the different Multiframes. The respective frame type is also indicated in the lower part of the table.

| | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 |
|---|---|---|---|---|---|---|---|---|---|
| TCH/F | ■ | | | | | | | ■ | ■ |
| TCH/H | | ■ | ■ | | | | | | |
| TCH/H | | | ■ | | | | | | |
| BCCH | | | | ■ | ■ | ■ | | | |
| FCCH | | | | ■ | ■ | | | | |
| SCH | | | | ■ | ■ | | | | |
| CCCH | | | | ■ | ■ | ■ | | | |
| SDCCH | | | | | ■ | | ■ | | |
| SACCH | ■ | ■ | ■ | | ■ | | ■ | ■ | ■ |
| FACCH | ■ | ■ | ■ | | | | | ■ | |
| Multiframe Type | 26 | 26 | 26 | 51 | 51 | 51 | 51 | 26 | 26 |

Table A.2.: Possible mappings of channels onto Multiframes

# Appendix B.

# OsmocomBB

This section contains general information about how to operate and setup the OsmocomBB framework and the Motorola C123.

## B.1. Installation

The environment used for this project was a Thinkpad X220 Tablet running Xubuntu Linux 11.10. The instructions should work for any other distribution of the Ubuntu product palette.

1. Build libraries must be installed on the operating system to enable compiling libraries.

   ```
   sudo apt-get install libtool shtool autoconf git-core
   pkg-config make gcc wget
   ```

2. The GNU Arm cross compiler toolchain needs to be installed so the firmware for the Motorola C123 can be built. It will be added as a repository to `sources` so it can be easily removed if it is not required any more.

   ```
   sudo add-apt-repository ppa:bdrung/bsprak
   sudo apt-get update
   sudo apt-get install arm-elf-toolchain
   ```

3. The source code needs to be obtained. This can be either done by checking out the latest version of the framework from the developers, or by using the code on the CD.

   ```
   git clone git://git.osmocom.org/osmocom-bb.git
   ```

4. At this point some firmwares had build errors, therefore we will compile only the firmware for the Calypso board used by the Motorola C123. This constraint might not be necessary if a newer version of the framework is used. In the `src` directory of the OsmocomBB framework the build process can be started.

```
make BOARDS=compal_e88
```

5. If a new version of OsmocomBB is used, the extra code from this project must be included in the build. The three files `catcher.c`, `app_catcher.c` and `pch_scan.c` must be moved to `osmocom-bb/src/host/layer23/src/misc` and the `Makefile.am` must be edited to include the new code.

```
bin_PROGRAMS = bcch_scan ... cbch_sniff catcher \
pch_scan
catcher_LDADD = $(LDADD) -lm
catcher_SOURCES = ../common/main.c app_catcher.c \
catcher.c ../../../gsmmap/geo.c
pch_scan_SOURCES = ../common/main.c pch_scan.c rslms.c
```

## B.2. Usage

To use a program written in the framework, the Motorola C123 needs to be flashed with the custom firmware. This can be done with the `osmocon` application.

```
cd src/host/osmocon

sudo ./osmocon -p /dev/ttyUSB0 -m c123xor
../../target/firmware/board/compal_e88/layer1.compalram.bin
```

After `osmocon` is started and running any application can be started with root privileges.

```
cd ../layer23/src/misc/
sudo ./catcher
```

The `pch_scan` program requires an ARFCN as an input. For example, to conduct a scan on the PCH of ARFCN 127 one would call:

```
sudo ./pch_scan -a 127
```

## B.3. Serial Cable Schematics

A T191 unlock cable used to connect the Motorola C123 can either be obtained by ordering it from one of the mentioned stores or by building it from scratch. These are the schematics required for building the unlock cable taken from a GSM Blog [1], which features images of many more cables for different brands.

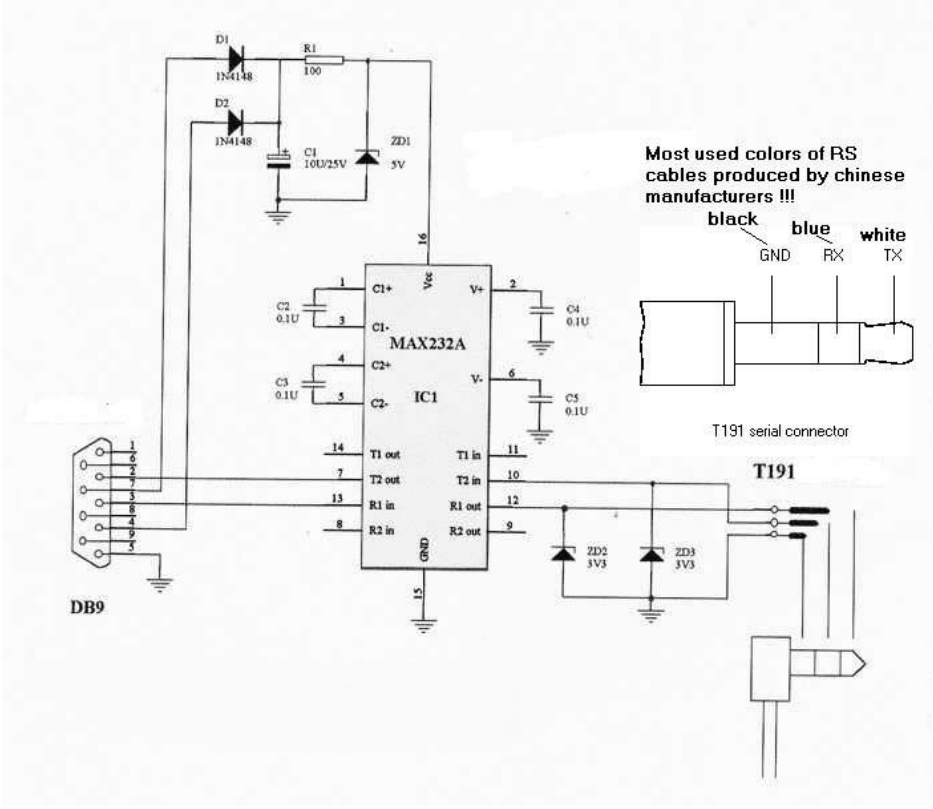



Figure B.1.: Serial cable schematics.

[1] GSM Box Schematics, `http://gsmringtonefree.blogspot.de/` [Online; Accessed 05.2012]

# Appendix C.

# IMSI Catcher Detection System

This section will cover some code related topics of the ICDS.

## C.1. Extextions

Rules, evaluators and filters are implemented in a way that new modules can be added quickly by way of inheritance and instantiating them in the constructor of the controller so they are known to the system. The following example shows how to implement a new rule and add it to the system. This exemplary process is nearly the same for filters and evaluators.

At first this base class has to be derived.

```
class Rule:
    #set whether the rule should be used by the
    #controller
    is_active = False
    #string that will identify the rule in the report
    identifier = 'Rule'

    #the logic of the rule, will be called by controller
    def check(self, arfcn, base_station_list):
        return RuleResult.CRITICAL
```

The new rule class needs to override the check method to do something meaningful. The identifier should also be set to a proper value.

```
class MyRule (Rule):
    identifier = 'My own Rule'
    def check(self,arfcn, base_station_list):
        result = RuleResult.CRITICAL
        #do some logic here and set result
        return result
```

`arfcn` and `base_station_list` are given to the check method by the controller. The first parameter is the ARFCN of the base station to which the evaluation will be applied. The second one is a list of all the base stations with complete information as far as it has been obtained by the ICDS. After it has been implemented it can be instantiated and added to the list of active rules in the constructor of the controller.

```
class PyCatcherController:
    ...
    def __init__ (self):
        ...
        self.my_rule = MyRule()
        self.my_rule.is_active = True
        self._rules.add(self._my_rule)
        ...
```

## C.2. Example Configuration

This example configuration has been used for the evaluation in the Freiburg area.

```
#Core Configuration ------------------------------------

#Settings for the Motorola C123 .
Device_settings = { 'mobile_device' : '/dev/ttyUSB0',
                    'xor_type' : 'c123xor',
                    'firmware' : 'compal_e88',
                   }

#Location of the osmocom library.
Osmocon_lib = '''/home/tom/imsi-catcher-detection/Src/
osmolib/src'''

#Generates commands from location and device settings.
#Does normally not have to be edited.
Commands = {'osmocon_command' : [Osmocon_lib +
    '/host/osmocon/osmocon',
    '-p', Device_settings['mobile_device'],
    '-m', Device_settings['xor_type'],
    Osmocon_lib + '/target/firmware/board/'
    + Device_settings['firmware']
    + '/layer1.compalram.bin'],
    'scan_command' : [Osmocon_lib
    + '/host/layer23/src/misc/catcher'],
    'pch_command' : [Osmocon_lib
    + '/host/layer23/src/misc/pch_scan'],
}

#Rules Configuration ------------------------------------

#A list of providers that should be taken as legitimate.
Provider_list = ['T-Mobile', 'O2', 'Vodafone', 'E-Plus']



#----------Continues on next page----------------------
```

```
#Countries where the given providers have presence.
Provider_Country_list = {
    'T-Mobile':'Germany',
    'O2':'Germany',
    'Vodafone':'Germany',
    'E-Plus':'Germany'
}

#Comma separated list of LACs that can be observed in the
#given area.
LAC_mapping = {
    'T-Mobile' : [21014,21015],
    'O2' : [50945],
    'Vodafone' : [793],
    'E-Plus' : [138,588]
}

#Frequency intervals that are registered to the
#given providers.
ARFCN_mapping = {
    'T-Mobile' : [(13,49),(81, 102),(122,124),(587,611)],
    'O2' : [(0,0),(1000,1023),(637,723)],
    'Vodafone' : [(1,12),(50,80),(103,121),(725,751)],
    'E-Plus' : [(975,999),(777,863)]
}

#How much % the LAC of a base station can deviate from
#the median before throwing an error (range 0 to 1 where
#0 means no tolerance).
LAC_threshold = 0

#How much % the rx level is allowed to be away from the
#interval located in the Location Area Database
DB_RX_threshold = 0.05

#How much % the rx is allowed to change during the course
#of a scan.
CH_RX_threshold = 0.02

#-------------Continues on next page--------------------
```

```
#How many neighbours need to be discovered
Neighbours_threshold = 3

#How much Pagings per 10s are required to give an Ok
#rating
Pagings_per_10s_threshold = 20

#How many hopping assignments are required to give
#an Ok rating
Assignment_limit = 0

#PCH Parameters ----------------------------------------

#How often a failed PCH scan should retry
PCH_retries = 5

#Time the PCH is scanned during Operation in
#User Mode
USR_timeout = 15

#Evaluator Configuration -------------------------------

#This configuration separates the different groups of
#rules from one another.

Rule_Groups = [
    ['Provider Check', 'Country Provider Mapping',
     'ARFCN Mapping', 'LAC Mapping', 'Unique CellID'],
    ['LAC Median Deviation', 'Neighbourhood Structure',
    'Pure Neighbourhoods', 'Fully Discovered
    Neighbourhoods'],
    ['Local Area Database','CellID Database'],
    ['LAC Change Rule','rx Change Rule'],
    ['PCH Scan']
]

#-------------Continues on next page----------------------
```

```
#Database Configuration --------------------------------

#The API key for OpenCellID.
#Can be freely obtained by registering on the web site.
Open_Cell_ID_Key = 'd7a5bc3f21b44d4bf93d1ec2b3f83dc4'

#Path to the folder where databases should be saved to or
#loaded from. The ICDS will look in this folder if data-
#bases are available.
Database_path = '''/home/tom/imsi-catcher-detection/Src
/PyCatcher/Databases/'''
```

# Appendix D.

# System Information

The following pages contain parsed System Information Messages of type 1–4 for reference [17].

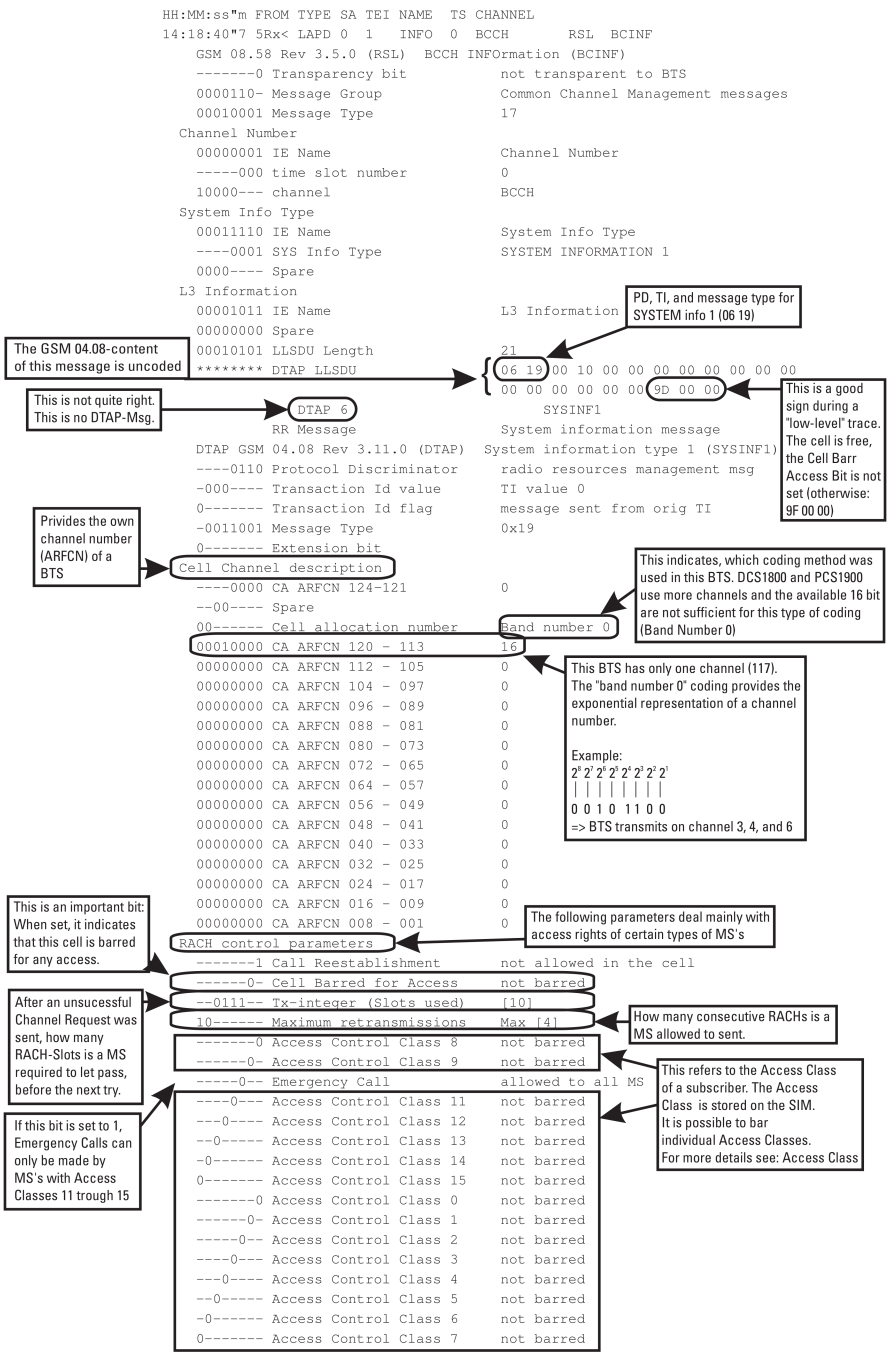



Figure D.1.: System Information 1 Message

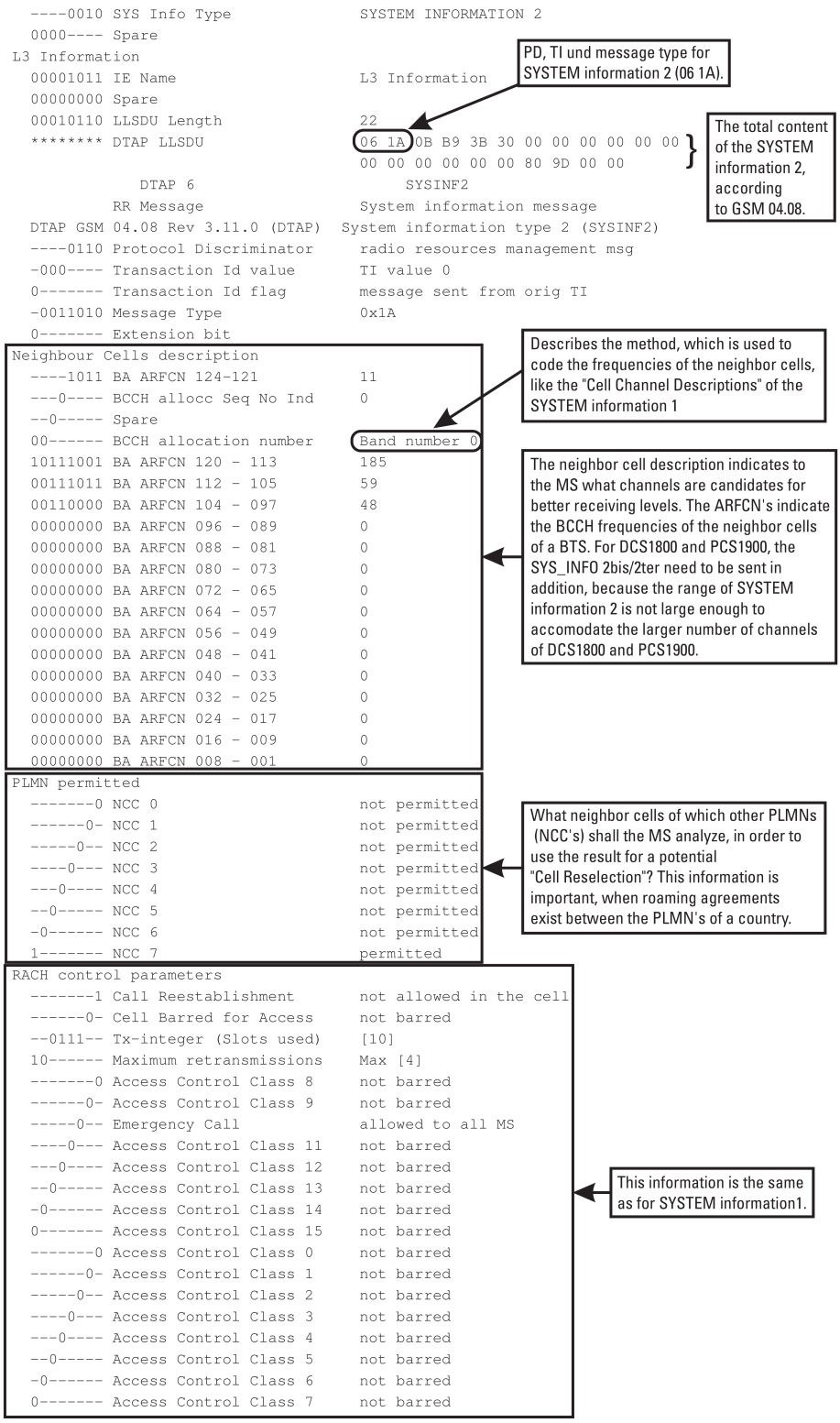





Figure D.2.: System Information 2 Message

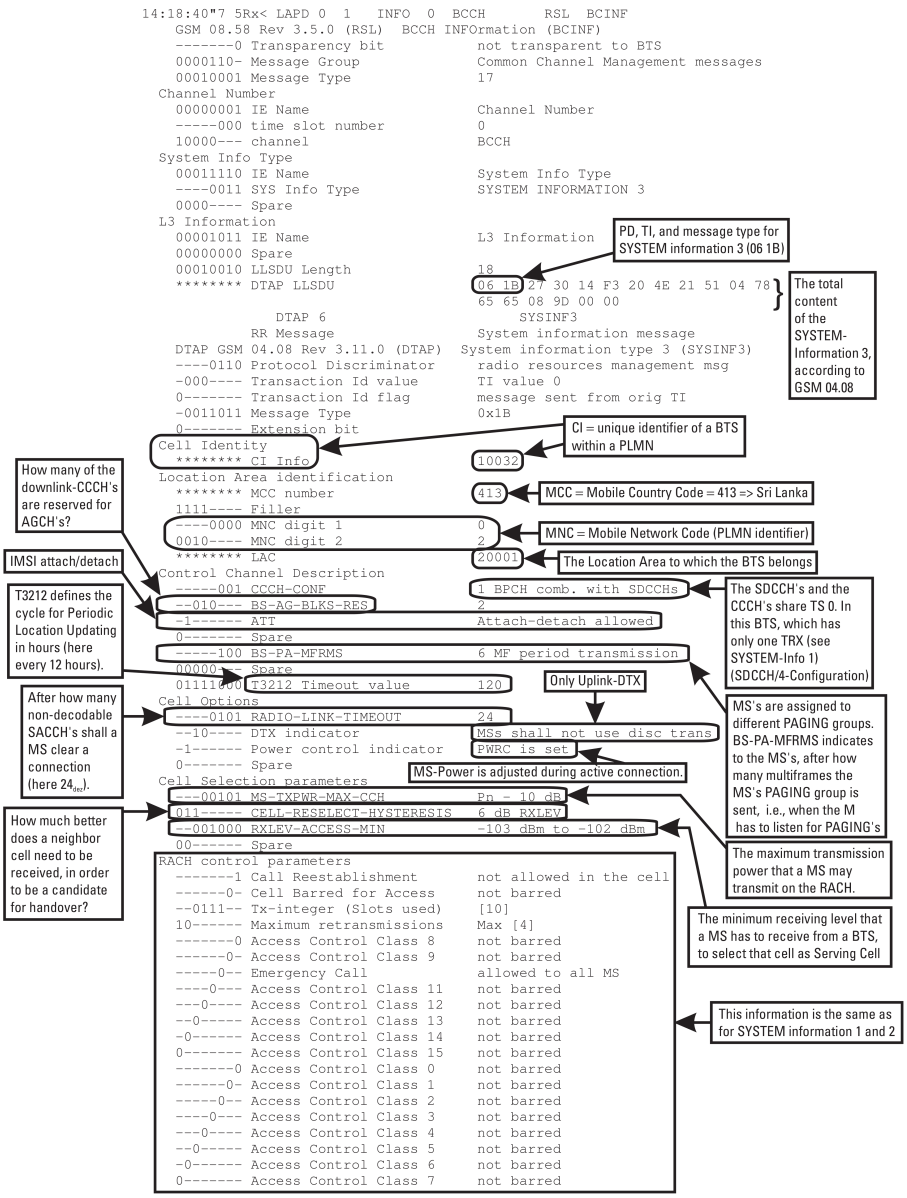



Figure D.3.: System Information 3 Message

```
14:18:40"7 5Rx< LAPD 0  1   INFO  0  BCCH       RSL  BCINF
    GSM 08.58 Rev 3.5.0 (RSL)  BCCH INFOrmation (BCINF)
    -------0 Transparency bit           not transparent to BTS
    0000110- Message Group              Common Channel Management messages
    00010001 Message Type               17
  Channel Number
    00000001 IE Name                    Channel Number
    -----000 time slot number           0
    10000--- channel                    BCCH
  System Info Type
    00011110 IE Name                    System Info Type
    ----0100 SYS Info Type              SYSTEM INFORMATION 4
    0000---- Spare
  L3 Information
    00001011 IE Name                    L3 Information
    00000000 Spare
    00001100 LLSDU Length               12
    ******** DTAP LLSDU                 06 1C 14 F3 20 4E 21 65 08 9D 00 00
                DTAP 6                      SYSINF4
             RR Message                 System information message
    DTAP GSM 04.08 Rev 3.11.0 (DTAP)  System information type 4 (SYSINF4)
    ----0110 Protocol Discriminator     radio resources management msg
    -000---- Transaction Id value      TI value 0
    0------- Transaction Id flag        message sent from orig TI
    -0011100 Message Type               0x1C
    0------- Extension bit
Location Area identification
    ******** MCC number                 413
    1111---- Filler
    ----0000 MNC digit 1               0
    0010---- MNC digit 2               2
    ******** LAC                       20001
Cell Selection parameters
    ---00101 MS-TXPWR-MAX-CCH          Pn - 10 dB
    011----- CELL-RESELECT-HYSTERESIS  6 dB RXLEV
    --001000 RXLEV-ACCESS-MIN          -103 dBm to -102 dBm
    00------ Spare
RACH control parameters
    -------1 Call Reestablishment      not allowed in the cell
    ------0- Cell Barred for Access    not barred
    --0111-- Tx-integer (Slots used)   [10]
    10------ Maximum retransmissions   Max [4]
    -------0 Access Control Class 8    not barred
    ------0- Access Control Class 9    not barred
    -----0-- Emergency Call            allowed to all MS
    ----0--- Access Control Class 11   not barred
    ---0---- Access Control Class 12   not barred
    --0----- Access Control Class 13   not barred
    -0------ Access Control Class 14   not barred
    0------- Access Control Class 15   not barred
    -------0 Access Control Class 0    not barred
    ------0- Access Control Class 1    not barred
    -----0-- Access Control Class 2    not barred
    ----0--- Access Control Class 3    not barred
    ---0---- Access Control Class 4    not barred
    --0----- Access Control Class 5    not barred
    -0------ Access Control Class 6    not barred
    0------- Access Control Class 7    not barred
```

PD, TI and message type for SYSTEM information 4 (06 1C).

The total content of the SYSTEM information 4, according to GSM 04.08.

This information on the Location Area is the same as in SYSTEM information 3.

This information on the Cell Selection Parameters is the same as in SYSTEM information 3.

This information is the same as for SYSTEM information 1, 2, and 3

Figure D.4.: System Information 4 Message

# Appendix E.

# Evaluation Data

## E.1. Rx and LAC Change Test

The following table contains the two configurations that have been used to test the LAC Change and rx Change Rules. Config 6 is identical to the configuration used on the base station and thus only triggers the rx Change Rule. Config 5 has a different LAC than the original base station and thus was used to test the former one. Additionally the rx Change Rule is also triggered for this configuration.

| | Config 5 | Config 6 |
|---|---|---|
| ARFCN | 877 | 877 |
| ShortName | 23 | 23 |
| MCC | 262 | 262 |
| MNC | 23 | 23 |
| LAC | 666 | 4711 |
| Cell ID | 1800 | 1800 |
| Neighbours | 806, 815, 817, 818, 823, 880 | 806, 815, 817, 818, 823, 880 |

Table E.1.: Configurations used for the rx / LAC Change Rules test.

## E.2. Database Rules Test

The following table contains the two configurations used to test the Database Rules. Config 6 is the same as before. It is used to check whether the Local Area Database Rule can find the difference in reception for the replaced base station. Config 7 features a new CID and is thus used to check if the Cell ID Database Rule is operating correctly.

| | Config 6 | Config 7 |
|---|---|---|
| ARFCN | 877 | 877 |
| ShortName | 23 | 23 |
| MCC | 262 | 262 |
| MNC | 23 | 23 |
| LAC | 4711 | 4711 |
| Cell ID | 1800 | 666 |
| Neighbours | 806, 815, 817, 818, 823, 880 | 806, 815, 817, 818, 823, 880 |

Table E.2.: Configurations used for the Database Rules test.

# Acronyms

| | |
|---|---|
| 3GPP | Third Generation Partnership Project 6, 7, 27 |
| AGCH | Access Grand Channel 18, 26 |
| ARFCN | Absolute Radio Frequency Number 16, 31, 40, 42, 45, 48, 49, 57, 58, 61, 69, 71, 74, 76 |
| ARIB | Association of Radio Industries and Businesses 6 |
| ATIS | Alliance for Telecommunications Industry Solutions 6 |
| AuC | Authentication Center 11, 14 |
| BCC | Base Station Color Code 42 |
| BCCH | Broadcast Channel 23, 25, 26, 39, 40, 42, 53, 66, 77, 79 |
| BGS | Bundesgrenzschutz 32 |
| BKA | Bundeskriminalamt 32 |
| BMI | Bundesministerium des Inneren 32 |
| BSC | Base Station Controller 15, 18, 19, 23, 27, 40 |
| BSIC | Base Station Identification Code 42 |
| BSS | Basestation Subsystem 7, 8, 15, 18 |
| BTS | Base Transceiver Station 15–21, 23, 27, 30, 31, 35, 39, 40, 42–46, 49, 52, 57, 58, 60, 61, 66, 74, 76, 79 |
| CC | Call Control 12 |
| CEPT | Conférence Européenne des Administrations des Postes et des Télécommunications 5 |
| CID | Cell Identity 16, 18, 28, 31, 42, 49, 50, 60, 73, 74, 76 |
| CSV | Comma Separated Value 57 |
| DCS1800 | Digital Cellular System 1800 5 |
| DIY | do-it-yourself 37 |
| DTMF | Dual Tone Multi Frequency 9 |
| EDGE | Enhanced Data Rates for GSM Evolution 7 |

| | |
|---|---|
| EEPROM | Electrically Erasable Programmable Read-Only Memory 10 |
| EIR | Equipment Identity Register 11 |
| ETSI | European Communication Standards Institute 5, 6 |
| FACCH | Fast Access Control Channel 19, 23, 25 |
| FCCH | Frequency Correction Channel 23, 25, 40 |
| FDMA | Frequency Division Multiple Access 20 |
| FN | Frame Number 40 |
| GMSK | Gaussian Minimum Shift Keying 26 |
| GPRS | General Packet Radio Service 7 |
| GSM | Global System for Mobile Communications 1–3, 5, 7, 9, 10, 12, 15, 16, 20, 27, 28, 30, 35–38, 46, 53, 65, 69, 77, 79, 80 |
| HDLC | High Level Data Link Control 26 |
| HLR | Home Location Register 11–14, 69 |
| HNI | Home Network Identifier 10 |
| HSDPA | High Speed Downlink Packet Access 7 |
| HSPA | High Speed Packet Access 7 |
| HSUPA | High Speed Uplink Packet Access 7 |
| IA | Immediate Assignment Message 18, 43, 51, 52, 57, 61, 68, 76, 79, 80 |
| ICDS | IMSI Catcher Detection System 2, 3, 25, 35, 38–40, 42–46, 48–55, 57, 58, 62, 65–69, 71, 73, 76, 77, 79, 80 |
| IMEI | International Mobile Equipment Identifier 9, 27, 30, 62 |
| IMSI | International Mobile Subscriber Identification 1, 10–14, 18, 27, 28, 30 |
| IN | Intelligent Network Subsystem 8 |
| ISDN | Integrated Services Digital Network 11, 27 |
| ITU | International Telecomunication Union 6, 11 |
| Kc | Ciphering Key 10, 14, 19 |
| Ki | Secret Key 10, 14, 19 |
| LA | Location Area 12, 13, 18, 46, 51, 68 |

| | |
|---|---|
| LAC | Location Area Code 28, 31, 42, 44, 50, 51, 60, 62, 68, 71, 74, 76, 79 |
| LAI | Location Area Identifier 28, 42 |
| LAPD | Link Access Procedure, D Channel 26, 27 |
| LAPD$_m$ | LAPD Mobile 27 |
| MCC | Mobile Country Code 10, 11, 42 |
| ME | Mobile Equipment 9, 18, 19, 36 |
| MM | Mobility Management 12 |
| MNC | Mobile Network Code 10, 11, 42 |
| MoU | Memorandum of Understanding 5 |
| MS | Mobile Station 7–9, 13–15, 18–21, 23, 25–28, 30, 31, 40, 42, 43, 46, 51, 52, 62, 68, 76, 79, 80 |
| MSC | Mobile Switching Center 11–14, 18, 19, 27 |
| MSIN | Mobile Subscriber Identification Number 10, 11 |
| MSISDN | Mobile Subscriber Integrated Services Digital Network Number 12, 13 |
| MSRN | Mobile Station Roaming Number 12 |
| MTP 2/SS7 | Message Transfer Part 2/SS7 27 |
| MVC | Model View Controller 53 |
| NCC | Network Color Code 42 |
| NMSI | National Mobile Subscriber Identity 11 |
| NMT | Northern Telecomunication 5 |
| NSS | Network Subsystem 7, 8, 11, 12, 19 |
| OMS | Operation and Maintainance Subsystem 8 |
| Osmocom | Open source mobile communications 35 |
| PCH | Paging Channel 18, 25, 26, 39, 42, 43, 51, 53, 57, 61, 62, 68, 77, 79, 80 |
| PIN | Personal Identification Number 10 |
| PLMN | Public Land Mobile Network 9, 11 |
| PSTN | Public Standard Telephone Network 7, 11, 12 |
| RACH | Random Access Channel 23, 24, 26 |
| RR | Radio Resource 27 |
| SACCH | Slow Access Control Channel 25 |
| SCH | Signalling Channel 21, 23, 25, 40 |
| SCP | Service Control Point 8 |

| | |
|---|---|
| SDCCH | Standalone Dedicated Control Channel 18, 19, 25, 26 |
| SF | Stealing Flag 23 |
| SIM | Subscriber Identity Module 9, 10, 14, 19, 28, 30, 32 |
| SMS | Short Message Service 9 |
| SS-7 | Signaling System 7 26, 27 |
| STC | Sub Technical Committee 5 |
| TACS | Total Access Communication System 5 |
| TC | Type Code 40 |
| TCH | Traffic Channel 19, 20, 25 |
| TDMA | Time Division Multiple Access 20, 21, 23, 24 |
| TMSI | Temporary IMSI 13, 18, 25, 43, 68, 69, 79 |
| TRAU | Transcoding Rate and Adaption Unit 15, 69 |
| TTA | Telecommunications Technology Association 6 |
| TTC | Telecommunications Technology Committee 6 |
| UMTS | Universal Mobile Telecomunications System 7, 14, 15, 80 |
| USRP | Universal Software Radio Peripheral 69 |
| VAS | value-added service 8 |
| VLR | Visitor Location Register 11–13, 69 |
| VoIP | Voice over IP 69 |