NTNU

Norwegian University of
Science and Technology

Mobile Network Security Experiments With USRP

Andre Mruz

Master of Science in Communication Technology
Submission date: June 2016
Supervisor: Stig Frode Mjølsnes, ITEM

Norwegian University of Science and Technology
Department of Telematics

Title: Mobile Network Security Experiments With USRP

Student: André Mruz

The International Mobile Subscriber Identity(IMSI) uniquely identifies a mobile network user subscription. This IMSI is used in the mobile network access control. A Subscriber Identity Module (SIM) contains and protects the IMSI and its associated cryptographic key in the user equipment. Some mechanisms are available for protecting the user's privacy, including a temporary identity scheme. This master thesis will investigate attacks on the privacy mechanisms in current mobile networks, by setting up and experiment with IMSI catchers. The purpose is to understand the behaviour and operation of IMSI catchers, then use this knowledge to propose efficient IMSI catcher detection.

More specifically, the first part is to build and experiment with IMSI catching based on the OpenBTS software and Universal Software Radio Peripheral(USRP) devices. The operation required of an efficient GSM IMSI catcher will be described. The second part aims to reproduce, and possibly extend, the privacy attacks on the 4G Long-Term Evolution(LTE) mobile system, as reported in [Sea15]. Here the experiments can be based on the Eurecom OpenAirInterface software platform which implements the LTE protocols.

Finally, if time allows, propose detection methods of IMSI catcher and false base stations .

Responsible professor: Stig Frode Mjølsnes, ITEM

Abstract

Though the mobile phone market has embraced 3G and 4G services, especially for the convenience of high speed Internet connectivity, the original cellular network system, GSM, is still available and covers most of the human population. One of its weaknesses is the relative simplicity of setting up a fake Base Transceiver Station (BTS), which can mount several attacks against Mobile Station (MS). One of them is to trick the MS into revealing International Mobile Subscriber Identity(IMSI), which uniquely identifies a mobile network user subscription. A breach of privacy. These devices have therefore become known as IMSI catchers.

This thesis presents IMSI catchers, how they can operate in the GSM network, and what makes them most effective at their task. This information is then used to identify ways detecting IMSI catchers. Some of these identification methods are implemented in a piece of software that utilizes a Universal Software Radio Peripheral (USRP) to pick up GSM signals. This IMSI Catcher Detector is then tested in a live GSM environment, and the results of this experiment are analyzed.

One proposed way of detection is to gather information from many BTSs to form a "fingerprint" of what is normal operation. Deviation from this would be deemed suspicious. This method seems to have some viability. After analyzing the data from the experiment it was found that there was very little deviation in broadcast traffic of the BTSs detected in the area covered by the experiment.

Sammendrag

Selv om mobiltelefonmarkedet har mottatt 3G of 4G svært godt, spesielt den høye internetthastigheten, er det originale mobilnettet, GSM, fortstll tilgjengelig, og dekker mesteparten av menneskeheten. En av dens svakheter er at det er relativt enkelt å sette opp et falsk bastetårn (BTS), som kan gjenomføre flere angrep mot mobiltelefonen (MS). Et av dem er å lure mobiltelefonen til å gi fra seg sin International Mobile Subscriber Identity(IMSI), som identifiserer en netverksabonnement. Et brudd på personvernet. Disse innretningene har derfor fått navnet IMSI catchers.

Denne avhanglingen presenterer IMSI catchers, hvordan de fungerer i GSM netverket, og hvordan man gjør dem mest effektive. Denne informasjonen blir så brukt til å identifisere hvordan man kan detektere en IMSI catcher. Noen av disse metodene blir så implementert i et program som bruker en Universal Software Radio Peripheral (USRP) enhet til å plukke of GSM signaler. Denne IMSI catcher detektoren blir deretter testet i et ekte GSM system, og resultatene fra dette eksperimentet blir analysert.

En av de foreslåtte deteksjonsmetodene er å samle informasjon om en rekke BTSer for å lage et "fingeravtrykk" av normal nettverksoppførsel.Avvik fra denne normalen blir sett på som mistenkelig. Denne metoden har muligens meritter. Etter å ha analysert resultatene fra ekspermentet, viste det seg at det er svært lite forskjell mellom meldingene kringkastet av BTSene i eksperimentområdet.

Preface

This is a master's thesis in information security written at the Department of Telematics (ITEM) at the Norwegian University of Science and Technology (NTNU), in Trondheim, Norway. It is based on research and work done over the course of the 2016 spring semester, the final one in the 5 year Communication Technology Programme, under the supervision of professor Stig Frode Mjølsnes.

I would like to thank all those who have been of assistance over the course of my work. My supervising professor, Stig Frode Mjølsnes for guidance and assistance. Fellow students Niklas Molnes Hole, Eirik Fosser and Jonathan Hansen for assisting hardware and software configuration, and debugging. Anyone who has who has contributed to the OpenBTS or GR-GSM projects for their hard work, these pieces of software that have been very useful during this project.

Trondheim, 2016 André Mruz

Contents

Appendices

List of Figures

List of Tables

List of Algorithms

List of Symbols

dBm	Decibel-milliwatts.
dBm	Decibel.
$=$	Equal.
$!=$	Not Equal.
$Fu(n)$	Downlin Frequency.
$Fl(n)$	Uplink Frequency.
kHz	Kilohertz.
MHz	Megahertz.

List of Acronyms

2G Second Generation of Mobile Telecommunications Technology.

3G Third Generation of Mobile Telecommunications Technology.

3GPP Third Generation Partnership Project.

4G Fourth Generation of Mobile Telecommunications Technology.

agch Access Grant Control Channel.

ARFCN Absolute Radio-Frequency Channel Number.

AuC Authentication Center.

BA BCCH Allocation.

BCCH Broadcast Control Channel.

BCH Broadcast Channels.

BSC Base Station Controller.

BSIC Base Station Identity Code.

BTS Base Transceiver Station.

CBCH Cell Broadcast Channel.

CCCH Common Control Channels.

CCH Common Channels.

CEPT Conférence Européenne des Postes et Télécommunication.

CI Cell Identity.

CRH CELL-RESELECT-HYSTERESIS.

CRO CELL-RESELECT-OFFSET.

DC Direct Current.

DCCH Dedicated Control Channels.

DCH Dedicated Channels.

DNS Domain Name System.

DOS Denial Of Service.

EIR Equipment Identity Register.

FACCH Fast Associated Control Channel.

FCCH Frequency Correction Channel.

FDMA Frequency-Division Multiple Access.

GPGGA Global Positioning System Fix Data.

GPS Global Positioning System.

GPSDO Global Positioning System Disciplined Oscillator.

GSM Global System for Mobile Communications.

GSMA GSM Association.

GUI Graphical User Interface.

HLR Home Location Register.

HN Home Network.

IMEI International Mobile Station Equipment Identity.

IMSI International Mobile Subscriber Identity.

IP Internet Protocol.

ISDN Integrated Services Digital Network.

ITEM Department of Telematics.

LA Location Area.

LAC Location Area Code.

LAI Location Area Identity.

LTE Long-Term Evolution.

LUR Location Update Request.

MCC Mobile Country Code.

ME Mobile Equipment.

MIC Message Integrity Code.

MNC Mobile Network Code.

MS Mobile Station.

MSC Mobile-services Switching Center.

MSIN Mobile Subscriber Identity Number.

MSISDN Mobile Station International ISDN Number.

NMEA National Marine Electronics Association.

NTNU Norwegian University of Science and Technology.

OS Operating System.

PCH Paging Channel.

PLMN Public Land Mobile Network.

PSTN Public Switched Telephone Network.

PT Penalty-Time.

RACH Random Access Channel.

RF Radio Frequency.

RLA Received Level Average.

Rx Reception.

RxL Received Signal Strength.

SACCH Slow Associated Control Channel.

SCH Synchronization Channel.

SDCCH Stand-alone Dedicated Control Channels.

SDR Software Defined Radio.

SI System Information.

SIM Subscriber Identity Module.

SINTEF The Foundation for Scientific and Industrial Research.

SIP Session Initiation Protocol.

SIPAuthServe SIP Authorization Server.

SMqueue SIP Message Queue.

SMS Short Message Service.

SN Serving Network.

SNR Serial Number.

SS7 Signalling System No. 7.

T3212 Periodic Location Updating Timer.

TCH Traffic Channel.

TCH/F Full Rate Traffic Channel.

TCH/H Half Rate Traffic Channel.

TDMA Time-Division Multiple Access.

TMSI Temporary Mobile Subscriber Identity.

TO Temporary-Offset.

TS Technical Specification.

TX Transmission.

UDP User Datagram Protocol.

UHD USRP Hardware Driver.

UMTS Universal Mobile Telecommunications System.

USIM Universal Subscriber Identity Module.

USRP Universal Software Radio peripheral.

VLR Visitor Location Register.

VM Virtual Machine.

VoIP Voice over IP.

Chapter 1
Introduction

GSM began its development in 1982. at that point a collaboration between representatives from Denmark, Finland, Sweden, Norway and the Netherlands[Aud08]. They met in june that year, in Vienna, and created a document that would serve as the baseline for the GSM system, GSM 2/82[dPeT82]. It would take ten years of further development before the system first came into service in 1992. Even at that time, several weaknesses were known[FHMN10, Aud08]:

- Cryptographic keys were short enough to eventually become vulnerable to brute force attacks.
- Parts of the security architecture(chiefly the cryptographic algorithms) were kept confidential.
- Encryption is not used in some networks.
- There was no authentication of the network to the mobile station.

The final of these will be the main subject of this paper. No network authentication opens up for potential attackers to spoof network equipment to the Mobile Station (MS)-and by extension the end user-and through this equipment gain access to location information and traffic from the MS. When the system was first designed, the cost of making such equipment was thought to be prohibitively expensive, and the risk was thought to be acceptable[FHMN10]. This is no longer the case. Professional, and purpose built equipment has been available for over a decade–add reference–, and anyone can get a simple version with Software Defined Radio (SDR) hardware priced similarly to that of a laptop and free software—add reference. Even though it can have more functionality, this sort of equipment is best known as an IMSI catcher.

Cellular networks have evolved since 1992. With the release of Universal Mobile Telecommunications System (UMTS), also called Third Generation of Mobile

Telecommunications Technology (3G), in 2002[NN03], and Long-Term Evolution (LTE), also called Fourth Generation of Mobile Telecommunications Technology (4G), in 2009, new and improved security protocols were introduced. However, the Second Generation of Mobile Telecommunications Technology (2G) GSM system is still widely utilized across the entire world, often as a backup system, and is estimated to covers over 90% of the world's population[22].

1.1 Changes to Problem Description

The original problem description mentions security testing with regards to the 4G system. However, work on detection mechanisms for IMSI catchers grew naturally from the first part of the project which studied this kind of equipment spoofing. It was eventually concluded that there was not enough time left to perform any meaningful work in 4G, so it was dropped in favor of focusing on detecting IMSI catchers in 2G.

The IMSI catcher detector developed in the thesis did not use any code from OpenBTS[Net16]. Instead, code from the open source project gr-gsm[Kea16] was customized, and used in the code base.

1.2 Methodology

The methodology used in this thesis can be divided into three parts. The first is research into cellular networks, IMSI catchers, and the detection of said IMSI catcher, through a literature study of relevant papers and textbooks.

The second is the development of two pieces of software. One which automatically configures an IMSI catcher, in the form of a USRP device running OpenBTS[Net16]. The other endeavors to detect IMSI catchers by implementing some of the detection methods found during the literature study.

The third is an experiment using the newly created IMSI catcher detector and the OpenBTS IMSI catcher. It aims to test the viability of the implemented detection methods, and to gain information about real life deployment of GSM.

1.3 Structure

This report consists of 6 chapters, including this one.

Chapter 2 presents GSM in enough detail to understand the following chapters.

Chapter 3 presents IMSI cathcers, how they can operate, be configured, and potentially be detected.

Chapter 4 presents a practical IMSI catcher detector that utilizes a USRP device to catch GSM traffic, and runs a series of tests looking for the behavior of an IMSI catcher. It also describes a piece of software that can automatically configure a fake Base Transceiver Station (BTS)(in other words an IMSI catcher) based on its surroundings

Chapter 5 presents a practical experiment using the previously described IMSI catcher detector, and an IMSI catcher implemented wit USRP and OpenBTS.

Chapter 6 presents the conclusions drawn from the information gathered, and work done in this report.

Chapter 2
GSM

This chapter gives an overview of the GSM system. It is by no means complete, but serves to give the reader enough knowledge to understand the following chapters. On some topics it is therefore brief, while on others thorough. These later ones are of chief importance to the topic IMSI Catchers, and the detection of such equipment. This information has been gathered from various books on the subject, like [Aud08, FHMN10] and publicly available specifications from the Third Generation Partnership Project (3GPP) website[Hom]

The chapter starts with an overview of how the network is structured, what entities it contains, and what their tasks are. After which it will cover channels and frequencies, location management, cell selection and security.

2.1 Topology

Figure 2.1 shows the topology of cellular networks, also known as the Public Land Mobile Network (PLMN), and is the basis of the coordinates used inside the network. The system is divided by country, and within each country it is further divided up into networks, which are are owned by that countries operators. Within a network, the operator will further divide it into location areas of related network entities. The country, network and location area are respectively identified by a three digit Mobile Country Code (MCC), a two or three digit Mobile Network Code (MNC), and a 16 bit Location Area Code (LAC). When put together in the form MCC/MNC/LAC, they form a Location Area Identity (LAI), which is the globally unique identity of a location area. With this information, a MS is aware of where it is located[3GP14, Aud08]. Finally, the location area is itself divided into cells, each of which is given a Cell Identity (CI), a unique two byte value in that location area.

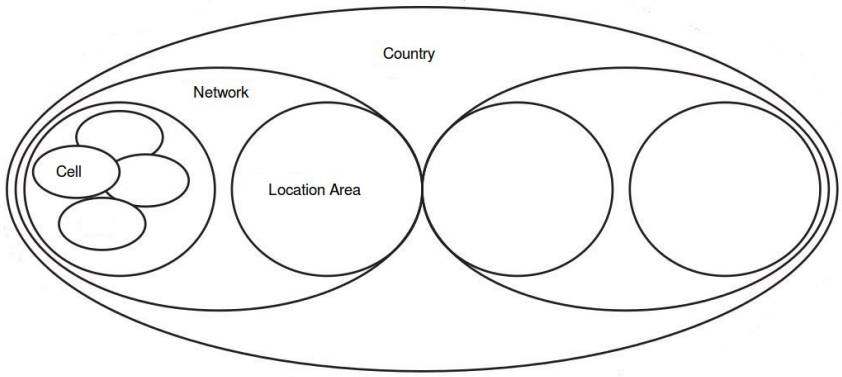

Figure 2.1: Cellular network topology, from [Aud08] and slightly altered

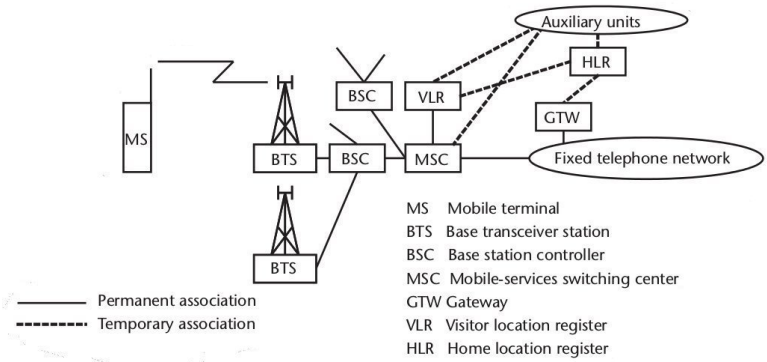

Figure 2.2: The GSM network entities[Aud08].

2.2 Network Architecture

An overview of the GSM entities can be seen in Figure 2.2. This section presents each of them, and their function within the system.

2.2.1 Mobile Station and Subscriber Identity Module

The MS is a telephone or other physical equipment which is able to connect to the GSM system over its radio interface, together with the software running on it. Its direct connection to the GSM system is with the BTS. It can be divided into two parts. First the Mobile Equipment (ME), the device itself, and second the Subscriber Identity Module (SIM), a token which contains the subscriber information[Aud08].

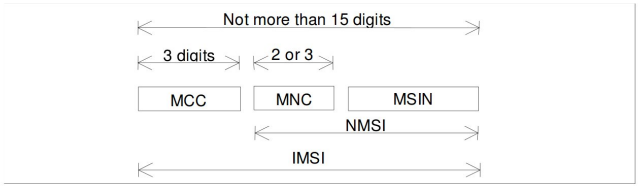

Figure 2.3: IMSI structure[3GP14].

The SIM is a small smart card inserted into the ME and contains the cryptographic key K_i, which is used for both encryption and authentication[FHMN10], see section 2.4, and the globally unique identifier IMSI. With the IMSI, the MS can identify itself to the network. Since the IMSI is tied to its owner, and the network needs to keep track of where the MS is located, the IMSI was identified as a potential source for privacy issues. If someone could keep track of the IMSI, they could keep track of its owner. Therefore, the GSM tries to limit the use of the actual value and the Visitor Location Register (VLR) assigns a temporary identifier to each MS, called a Temporary Mobile Subscriber Identity (TMSI).

The IMSI consists of thre parts: MCC, MNC, and Mobile Subscriber Identity Number (MSIN). It can be seen in Figure 2.3

Even though the MS and SIM technically separate entities, they are closely connected. In the rest of this text, unless otherwise specified, when a MS is mentioned, it is assumed to have a functioning SIM.

2.2.2 Base Transceiver Station

A BTS implements the GSM radio interface-called the Um interface-and is thus the moving MS's connection point to the PLMN, through which communication with the MS travels. Its second interface is with the Base Station Controller (BSC), on an interface called Abis. Of these interfaces, only the Um is encrypted[FHMN10].

Each BTS operates a single cell, and can therefore be identified by its CI. However, it is not limited to only one frequency. The operator is free to choose how many of the 124 possible frequencies each cell shall use, in fact the protocol sets no limit[3GP09], however it needs to make sure that the frequencies of overlapping of cells are not equal, to insure noninterference. For more information on the wireless channels, see section 2.3[Aud08].

2.2.3 Base Station Controller

A BSC is in charge of controlling one, or potentially more, BTSs across the Abis interface. It is primarily has the task of organizing frequency usage, signal management, and access management for its connected base stations. The BSC will also be connected to a Mobile-services Switching Center (MSC) over the A interface[Aud08].

2.2.4 Mobile-services Switching Center

The MSC is the GSM systems gate to the rest of the Public Switched Telephone Network (PSTN). It can be connected to several BSCs across the A interface, and takes care of call switching, and handover. Upon handover between BTSs, whether the BTSs are connected to the same BSC, the same MSC, or separate MSCs,the MSC is always the responsible entity. It is also connected to the VLR, and Home Location Register (HLR), by the B and C interfaces, respectively[Aud08].

2.2.5 Visiting Location Register

The VLR is in charge of mobility management connected to one or more MSCs over the B interface. It is a database which stores subscriber information such as telephone number, IMSI, authentication information, and the HLR in which the MS is registered, for all MSs/SIMs that are connected to any of its MSCs. This information is sent from the MS's HLR in its home network. Among other things, it is a vital entity in authentication and location update procedures, see sections 2.5 and 2.4[Aud08, Pag02].

2.2.6 Home Location Register

Located in the MS's home network, the HLR is a subscription database, similar to the VLR. However, subscription information is permanently stored in the HLR, and is the source from which the VLR gets its information. Together with the users subscription information, the HLR also stores the location of the MS, the MCC, MNC, and LAC, and uses this information to route calls for the MS to the correct location[Aud08].

2.2.7 Authentication Center

The Authentication Center (AuC) is a part of the HLR, but is often treated as a separate entity. It stores the secret key K_i, a shared secret with the SIM. This key is used for authenticating the MS to the network, and cipher key generation for encrypting radio traffic between MS and BTS [Aud08, Pag02].

2.3 Channels

This section presents the channels used by the Um interface between MS and BTS. It is separated into two sections, section 2.3.1 presents the physical layer of transmitting information, while section 2.3.2 presents the logical separation of information sent over that channel

2.3.1 Physical Channels

GSM uses a combination of Frequency-Division Multiple Access (FDMA) and Time-Division Multiple Access (TDMA). Different versions of GSM wil use different frequency bands(GSM-900, GSM-480, T-GSM-360,...), however all versions separate each carrier signal by 200. Equations 2.1 and 2.2[3GP16d] show the calculations for uplink and downlink frequencies in the GSM-900 band.

$$Fl(n) = 890 + 0.2 * n \tag{2.1}$$

$$Fu(n) = Fl(n) + 45 \tag{2.2}$$

In these equations, Fl(n) is the uplink frequency, Fu(n) is the downlink frequency, and n represents the Absolute Radio-Frequency Channel Number (ARFCN), which is the combination of downlink and uplink frequencies used by a BTS. Network operators must take care to plan their networks so that neighboring or overlapping cells do not share the same ARFCN. The TDMA scheme further divides each signal into eight equaly long timeslots. Together, eight of these timeslots are known as a frame[3GP16b, 3GP16d, Aud08].

2.3.2 Logical Channels

The information sent over the physical channels are divided into logical channels. These channels can first be divided into two groups, the Common Channels (CCH) and Dedicated Channels (DCH)[3GP16c].

The DCH is further divided into two groups: Dedicated Control Channels (DCCH) and Traffic Channel (TCH). The TCH are supposed to carry speech signals, there are two types: Full Rate Traffic Channel (TCH/F) and Half Rate Traffic Channel (TCH/H). Three channels are a part of the DCCH. They are Stand-alone Dedicated Control Channels (SDCCH), Slow Associated Control Channel (SACCH), and Broadcast Control Channel (BCCH). SDCCH is used for location updates, authentication, and call setup. SACCH is used to send power and timing information. BCCH is used for authentication and handover procedures[3GP16c].

Table 2.1: System Information messages 1 through 4[3GP09].

Type	Description
SI 1	Random access parameters to access system, and ARFCNs used by the cell.
SI 2	The BCCH Allocation (BA) list of neighboring frequencies/ARFCNs, and PLMN info.
SI 3	LAI, CI, CELL-RESELECT-OFFSET (CRO), CELL-RESELECT-HYSTERESIS (CRH), Periodic Location Updating Timer (T3212), and other cell selection parameters. Paging group information. Performance Options.
SI 4	LAI. Cell selection parameters. RACH control information.

The CCH is also divided into two groups: Common Control Channels (CCCH) and Broadcast Channels (BCH). The BCH consists of Frequency Correction Channel (FCCH), BCCH, and Synchronization Channel (SCH). The BCCH broadcasts information to the MS about the cell, the SCH helps the MS synchronize with the TDMA scheme, while the FCCH helps the MS synchronize to the frequency of the base tower.

The final four channels are in the CCCH: Access Grant Control Channel (agch), Random Access Channel (RACH), Paging Channel (PCH), Cell Broadcast Channel (CBCH). agch grants the MS a DCCH. the RACH is used by the MS to request access to the BTS. PCH is used for paging. CBCH broadcasts Short Message Services (SMSs), meaning that they are received by all MSs currently in the cell[3GP16c].

System Information (SI) messages are sent over the BCCH and SACCH. They used to facilitate MS operation with the cellular network by informing it about the BTS. Because these messages contain so much information about the BTS and cell, they are an important part of trying to detect fake BTSs. The 3GPP technical specification defines 24 different kinds of messages[3GP09], though not all need to be implemented for a network to function. Table 2.1 gives a short presentation of the first four.

2.4 Security

GSM aims to provide MS authentication, encryption, and user privacy. To achieve the first two the SIM and AuC have a shared secret key K_i. With this 126 bit key, the algorithm A8 creates the session key K_c, which in turn is used by the encryption algorithm A5 to encrypt data over the Um interface. The A3 algorithm is responsible

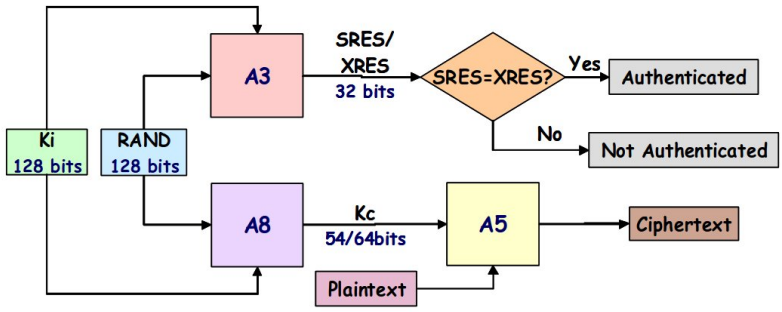

Figure 2.4: GSM cipher key generation, encryption, and authentication mechanism [Pag02].

for authentication. Figure 2.4 shows how these keys and algorithms are related [Pag02, FHMN10].

2.4.1 Authentication Procedure

Figure 2.5 shows a sequence diagram of the GSM authentication procedure.It is a challenge/response protocol. When the network requires a MS/SIM to authenticate itself, the VLR sends a Request Authentication message to the SIM, to which the SIM answers with a IMSI if this is the first authentication after booting up, or in most cases TMSI when the value has already been assigned. If this is the first time the SIM authenticates to the VLR, the database will try to connect to the previous VLR to get the IMSI. In the case that such a connection is not feasible, and the IMSI is still unkown, the VLR will request the IMSI from the SIM[FHMN10, Aud08].

The VLR connects to the HLR/AuC in the SIM's home network, and retrieves security parameters XRES, RAND, and K_c. The AuC has the SIM's K_i stored, and creates the parameters through the process described in section 2.4. The SIM now has to prove itself by correctly computing the SRES value. If XRES and SRES are equal, the authentication procedure has successfully completed [FHMN10].

Notice that the network never authenticates itself to the SIM.

2.4.2 Encryption

Encryption takes place over the Um radio interface. As shown in the two previous sections, the session key K_c is a byproduct of the authentication procedure and the A8 key generation algorithm A8, with k_i and RAND as input. K_c and RAND are sent to the visiting network, and the SIM receives the RAND, so that it too can calculate K_c

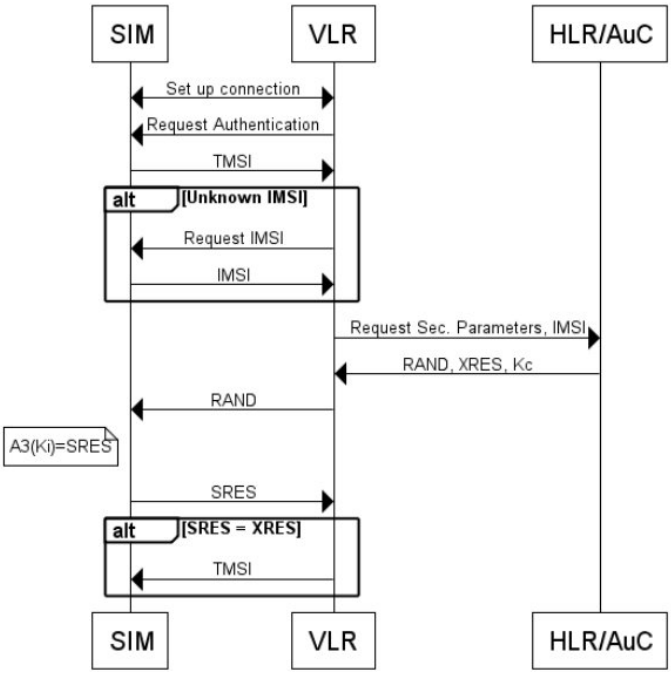

Figure 2.5: Authentication procedure[FHMN10].

Three stream ciphers have been standardised, A5/1, A5/2, and A5/3. When A5/1 and A5/2 were created they were criticized, because they were kept confidential. They have since the reverse engineered. In addition, A5/0 denotes the case when no encryption is applied. [FHMN10].

There have also been reports on breaking A5/1 over the years, like [NP09]. In GSM Associations (GSMAs)'s assessment of the situation [GSM09], the attacks would need to much ciphertext to be effective in practice. However, in a supposed document[bGG11] from the surveillance company GAMMA GROUP[Gro16], key recovery is described as possible and part of surveillance deployment. Though it does not go into detail on how this is achieved.

2.4.3 Privacy

GSM tries to achieve some user privacy by using the IMSI as little as possible, since it uniquely identifies the user in the network. This is achieved by assigning a temporary identity to each SIM/MS. This happens after completing the authentication procedure[Aud08, FHMN10].

2.5 Location Management

The core goal of cellular networks, and what separates them from landlines, is that the phone can initiate, receive, and keep a connection going while moving. In order to do this, the system need to keep track of the MS's location. The VLR keeps track of where MS in its location area are, and the HLR keeps track of which location area its MSs are currently in.[3GP99, 3GP16a].

2.5.1 Location Update

Location Update is the procedure to update the HLR about the MS's new location. As part of location update, the authentication procedure, described in section 2.4.1, is also performed. It has two triggers of specific note: Connecting to a cell in a different Location Area (LA) than the previous cell, this includes first connection after booting up, and when the T3212 runs out. T3212 is a constantly ticking down towards zero, with the initial value broadcast by the cell. When changing to a cell in the same LA but lower initial T3212 value, the timer is recalculated according to equation 2.3 [3GP99, Eve14].

$$Timer \equiv Timer \ mod \ T3212 \tag{2.3}$$

2.6 Idle Mode and Camping

Idle mode is the state a MS is in when it does not have a DCH, while camping on a cell is that state of being connected to a cell. When in this mode it still pursues two procedures: Cell selection and reselection. These procedures decide which cell the MS will camp on.

2.7 Cell Selection

The procedures for cell selection and reselection depend the two values: path loss criterion C1, and the reselection criterion C2. The following calculations are in dBm, and taken directly from 3GPP specification [3GP05].

Path loss criterion:

C1 = (A - Max(B, 0))

A = Received Level Average (RLA) - RXLEV_ACESS_MIN

B = MX_TXPWR_MAX_CCH - P

RXLEV_ACCESS_MIN = Minimum received signal level at the MS required for access to the system.

MS_TXPWR_MAX_CCH = Maximum TX power level an MS may use when accessing the system until otherwise commanded.

POWER OFFSET = The power offset to be used in conjunction with the MS_TXPWR_ MAX_CCH parameter by the class 3 DCS 1800 MS

P = Maximum RF output power of the MS

Reselection criterion:

For Penalty-Time (PT) != 11111:

C2 = C1 + CRO - Temporary-Offset (TO) * H(PT - T)

For PT = 11111:

C2 = C1 - CRO

For serving cells:

$H(x) = 0, x < 0$

$H(x) = 1, x \geq 0$

For non-serving cells:

H(x) = 0

2.7.1 Stored Cell Selection

The BA list of neighboring frequencies is broadcast from every cell. Stored cell selection is used when this list is available to the MS at startup. Only these frequencies are scanned. The six strongest signals are C1 calculated, and the highest

Table 2.2: Triggers for reselection. .

C1 falls below 0, for 5 seconds.
C2 of another cell is higher than that of the current cell for 5 seconds. If this cell is in another LA, C2 of the new cell must exceed the old C2 by the CRHdB of the old cell. If a reselect happened withing 15 seconds, the new C2 must exceed the old C2 by 5 dBm for 5 seconds.

is chosen as the cell the MS camps on. If this list is not available, normal cell selection is used.[3GP05].

2.7.2 Normal Cell Selection

Normal cell selection is used when the MS is unaware which type of GSM is currently in use. It will scan all frequencies and look for the six strongest cell signals. For these six, it will calculate C1, the highest becomes the cell which the MS camps on[3GP05].

2.7.3 Cell Reselection

After cell selection, the MS will keep looking for better cells to camp on. This is cell reselection. Like the stored cell selection procedure, it keeps track of the strongest frequencies from the BA neighbor list and continuously calculates C1 and C2. The following events cause the MS to reselect cell [3GP05].

If no suitable cell is for 10 seconds, the normal cell selection procedure is to take place.

Chapter 3
IMSI Catchers

This chapter describes IMSI catchers. The first section goes into detail about different ways in which they can operate, and how to make them do this effectively. The second describes different ways one may detect IMSI catchers.

3.1 Catching IMSIs

IMSI catchers take advantage of the fact that GSM does not authenticate the network devices to the MS. This has been known since the system was first designed[FHMN10], however the risk was thought to be acceptable, and further development was not warranted. In large part, this was because the equipment was prohibitively expensive to produce at that time. However, as time progressed equipment cost fell, and equipment created specifically to do this became available and specialized companies began to advertise this kind of service[Gro16]. Mostly to state organizations [EFF13].

The cost of equipment kept falling, and with the rise of SDR, it was shown that it was possible to create a low cost IMSI catcher at home [Pag10].

The IMSI uniquely identifies a mobile network user subscription, it is directly connected to the user. When a fake BTS picks up the value, it knows the user is nearby. In other words, the user's movement can be tracked to withing the radius of the IMSI catcher. A breach of privacy.

These devices send, receive, and behave like a normal BTSs would do. Since the MS has no way of authenticating the GSM network, if a fake BTS the most viable the the MS, it has no other option than to connect. Figure 3.1 shows the basic operation of an IMSI catcher. After deciding that the IMSI catcher is the mot suitable BTS, it connects and has to go trough the location update and authentication procedures. As opposed to regular networks, the IMSI catcher does not connect to another VLR or the home network. It will always request the IMSI, and the MS will comply[Pag10, vdBVdR15, HH14].

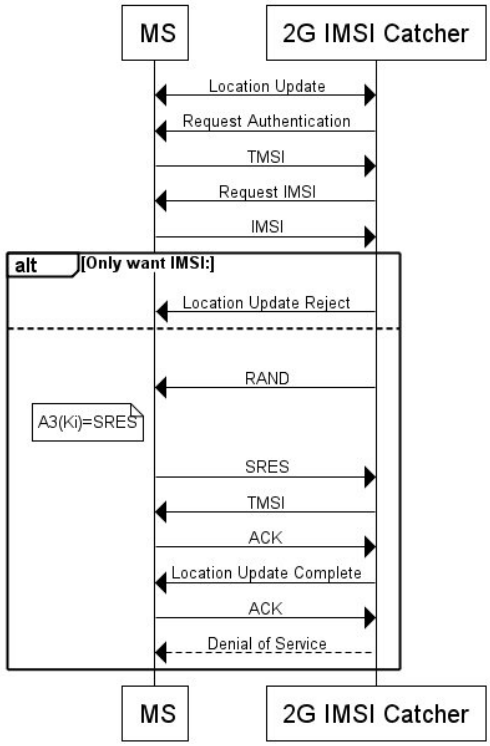

Figure 3.1: Catching IMSIs .

At this point, the IMSI catcher has a choice which depends of what its user wishes to achieve. If one only wants to track IMSIs, the IMSI catcher can send a Location Update Reject message, decoupling the MS and fake BTS. If not, the IMSI catcher can complete the location update procedure. The MS think it is connected to a legitimate BTS, but is in fact isolated from the GSM network. It can neither receive, nor initiate calls or SMS to other users of the network. Achieving a Denial Of Service (DOS) attack[HH14, Pag10].

The only calls of SMSs the MS would receive would be from the operator of the IMSI catcher. If it has these capabilities.

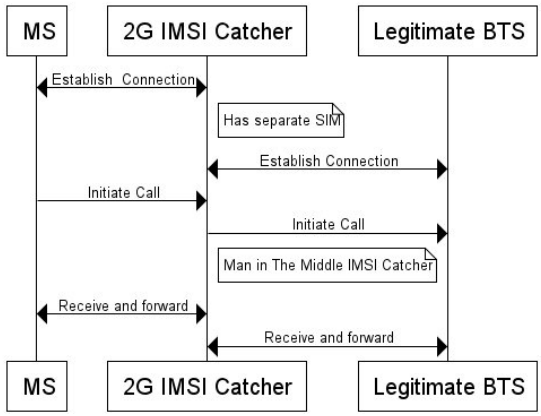

Figure 3.2: IMSI catcher with intercept ability.

A more advanced IMSI catcher will be able to extend the attack. By itself being a legitimate combination of MS and SIM, it could establish its own connection with the legitimate network. A simplified sequence diagram of this can be seen in figure 3.2[bGG11, HH14].

At this point the victim MS would be invisible to the network. Only the IMSI catcher/MS would be visible. Consequently other legitimate MSs would still not be able to call the victim MS. However, if the victim MS initiated a call of SMS, the IMSI catcher could create its own connection with the rest of the network and merely forward any information[HH14].

This extended attack would have to use A5/0 encryption, in other words no encryption. Even though the authentication procedure has been completed, the IMSI catcher is not aware of the K_i or K_c. And could therefore not encrypt or decrypt traffic to the victim MS.

A document[bGG11] which has been attributed to the GAMMA GROUP[Gro16], a professional surveillance company, claims it can attain the key as part of its services(though it does not specify how). In such a case one would not need to use A5/0.

3.1.1 Effective IMSI Catchers

If one wishes an IMSI catcher to operate optimally. One would want it to configure it to be the most likely candidate cell when a MS does its cell selection and reselection procedure, make sure the location update procedure is triggered, and if it mounts a DOS attack the MS shold stay connected to the IMSI catcher. Dabrowski et al.[ea14]

identifies such parameters.

- Use LAC different from nearby cells to cause location update procedure.
- Alternatively, have the same LAC but low T3212, causing a location update to be triggered when it reaches zero.
- Have a strong signal. This keeps the C1 value high. Increase chance of selection and reselection
- High CRO. Increases C2 value. Increase chance of reselection.
- Low TO increase C2.
- Low PT, keep H(x) function low.
- Low MX_TXPWR_MAX_CCH, to keep C1 at maximum
- Low RXLEV_ACCESS_MIN to keep C1 at maximum
- High CRH to lower chance of MS reselecting.
- BA list has only frequencies of weak or unused signals, reselect procedure will only get low C2 values.

3.1.2 3G and 4G downgrade

4G and 3G are also competitors of the IMSI catcher. In fact, many modern smart phones will prefer 4G or 3G over 2G. This is why one might wish stop these network from being operaitonal.

In [Sea15], Shaik et al., identifies a way of forcing the phone - called User Equipment (UE) in 3G and 4G - connect to 3G or 2G. This attack exploits that the in the Tracking Area Update (TAU) procedure, the "TAU Reject", sent from the network and causing the UE to disconnect from the network, does not require integrity protection. Thus removing the need for mutual authentication. Wetzel and Meyes similarly shows the "Location Update Reject" message in 3G is accepted without the need for mutual authentication in [WM15]. Without access to 3G or 4G the phone is forced to use GSM, where ofcourse the IMSI catches is waiting.

The combination of these downgrade attacks is suggested in [Dab16], and can be seen in figure 3.3.

Another way of forcing downgrade is to simply jam the 4G and 3G frequencies with a powerful noise generator. When the phone is unable to pick up the signal, it tries GSM, and normal IMSI catcher procedure is used [FHMN10].

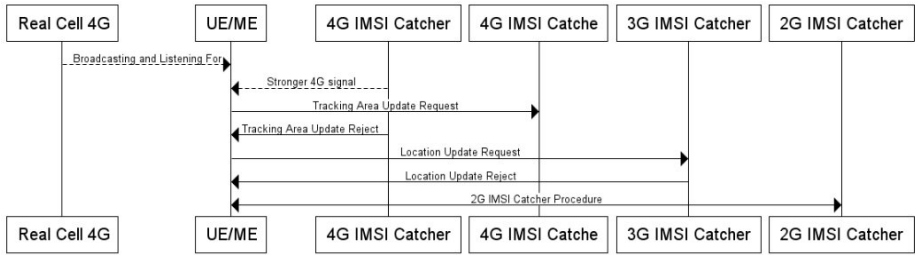

Figure 3.3: Combined downgrade of 4g & 3G, from [WM15, FHMN10] .

3.2 Detecting IMSI Catchers

This section present possible ways of detecting IMSI catchers.

3.2.1 Identifying Deviations and Suspect Traffic

Identifying suspect behavior starts with gathering traffic. One need to be able to pick up and process GSM signals. The *gr-gsm* software project[Kea16], for example, consists of several scripts for a number of SDRs to pick up GSM traffic. Even more impressive is the OsmocomBB project[OSM], which implements the entire GSM stack on a smart phone, but makes the internals available to the user and controllable from a laptop. Of course, nothing stops you from creating your own hardware or SDR code if you are capable enough.

Once the signal processing section takes care of, we must start to identify behavior which is synonymous with IMSI catcher deployment. The list in section 3.1.1 seems like the perfect starting point. Assuming that the operator of the fake BTS wants it to operate as effective as possible [vdBVdR15, ea14].

- Identify which LACs are used in an area. If one stands out as being surrounded by other LACs, it is suspicious.
- Alternatively, very low T3212 values are suspicious. The value of 1, would cause a location update after 6 minutes.
- Especially strong signals that appear in a small area could indicate a small local IMSI catcher
- Check for very high CRO values.
- Check for low TO values.
- Check for low PT values.

- Check for low MX_TXPWR_MAX_CCH values.
- Check for low RXLEV_ACCESS_MIN values.
- Check for high CRH values
- Check if the BA list has only frequencies of weak or are unused.

3.2.2 Network Fingerprinting

The previous section focused on values that were directly connected to GSM selection, reselection, authentication and location update procedures. In addition to this one could look at the larges picture. The amount of variables sent in all the different SI messages is staggering. Gather all of these, and on a network operator by operator basis find which variables stay the same. For variables that have some number value, one can calculate averages and standard deviations. Cells that deviate in many of these categories would stand out as particularly suspicious.

3.2.3 Active Tests

These tests requires the software running them to implement the full GSM stack, or at least enough to be able to send messages back and fourth between the network the the IMSI catcher detector.

The authentication procedure is suposed to authenticate the MS to the network. But what if the cold be used the other way around. An IMSI catcher that can not check with the legitimate network could verity the SRES values during authentication. Therefore, if one can configure a MS to send the wrong SRES value during the authentication procedure, and it is accepted, one could be reasonable sure it is an IMSI catcher. Especially if it is repeatable. Figure 3.4 shows how this test would be done.

In [Dab16], it is noted that a pure IMSI catcher, not interested in DOS attacks would most likely send a location update reject message every time the location update procedure was initiated. Similarly repeatedly TAU reject and 3G location update reject messages would indicate that 4G and 3G, respectively, are being blocked.

3.2.4 Known Information

All of these tests presumes that the tester known nothing about the network. The more knowledge a IMSI catcher detector has about the network, the more tests it could preform. The network operators themselves seem like the obvious choices for initiating such well informed test, but I have not found any information about such test. It is possible such test would be kept secret by the operators.

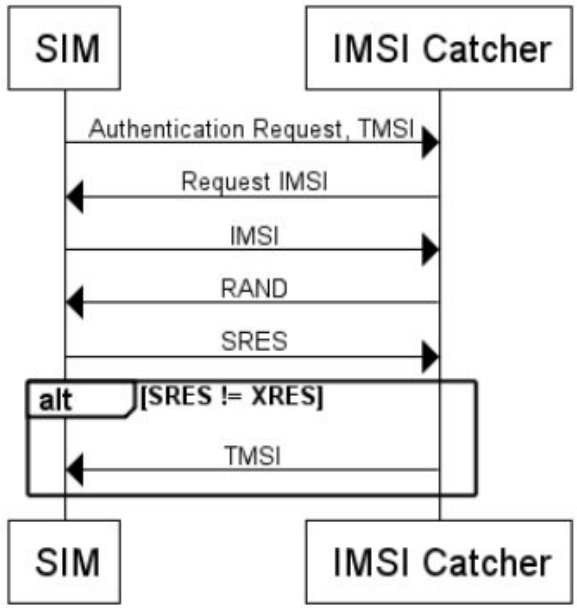

Figure 3.4: Bad Authentication.

Chapter 4
USRP Software

This chapter describes two pieces of software developed as part of this master's thesis. The first is a IMSI catcher detector, which scans GSM traffic looking for suspicious behaviour that would indicate a fake base tower. It utilizes some of the tests described in section 3.2.The second program auto configures an IMSI catcher.

They both utilize USRP hardware, a range of SDRs from Ettus Research[Res16a], to gather GSM traffic broadcast by all nearby BTSs, and then inspect this traffic. The software should work on any USRP device, but has only been tested with the Ettus N200. The programming language used is python[Fou16], though some underlying libraries are written in c++.

These libraries are a part of GNU Radio. To install them, you can follow the guide in appendix C.

The pseudo code shown throughout this chapter is a simplification, and is meant to help convey the essentials of certain parts of the code. It does not show full python implementation details. If you wish to see or try the code, it is freely available at a public github repository[Mru16].

In order for this code to be useful, one should have a machine running linux and a connected USRP device. The simplest way of connecting a networked USRP device to your is through a static Internet Protocol (IP) interface. Such a guide is available in appendix A. One should also configure some kernel parameters for the software to run optimally. A guide is available in appendix B

4.1 A USRP IMSI Catcher Detector

The script IMSI_Catcher_Catcher.py is an IMSI catcher detector that aims to detect fake base towers in GSM. Though it uses many of the tests described in section 3.2, it could not implement all of them, as it was limited by not having the code

base of of a full GSM MS stack and not enough time to develop such software. The program does tests based of BTS broadcast traffic, and knowledge of the network gained outside of the scans, if such knowledge is available.

The execution of the program can be divided into three parts: Setup, traffic gathering, and tests.

4.1.1 Setup

When first starting the program, the user will be asked to enter any information about the network he already has available. The user will be asked about the country's MCC, which MNCs are in use, and if there are any frequencies that are known to not be in use at that location. If this information is not entered, the execution will continue as normal, but the amount of tests are reduced accordingly.

The user is the asked if he wishes to get a Global Positioning System (GPS) location, or enter the location manually. The GPS functionality is only available if the USRP equipment has a GPSDO kit and antenna attached. This is implemented in code by an object that has a connection to the USRP device. Through this object we have access to two GPS related sensors: "gps_locked" and "gps_gpgga". We repeatedly query the gps_locked sensor until it returns True, then we get the sensors current position in the form of a Global Positioning System Fix Data (GPGGA) National Marine Electronics Association (NMEA) sentence from the gps_gpgga sensor.

Algorithm 4.1 Attaining GPS location in pseudo code

```
def get_gps_location() {
    usrp_device = gnuradio.uhd.usrp_source(addr="192.168.10.2");
    gps_lock = False;
    while (not gps_lock){
        gps_lock = usrp_device.get_mboard_sensor("gps_locked");
    }
    return usrp_device.get_mboard_sensor("gps_gpgga");
}
```

Next, the program will check if an xml logg file of previously scanned cells and frequencies is present. If so, its content is read and added to the programs set of previously scanned cells and frequencies.

The IMSI catcher detector is now ready to scan and check for suspicious behavior. Before a scan is initialized, the user enters a scan speed. This value can range from 0 to 5. Higher speed, means the scan finishes quicker, but at the risk of not detecting weaker signals. The scan can be done an arbitrary amount of times.

4.1.2 Gathering Traffic

For picking up GSM traffic, the program uses code from the open source project *gr-gsm* [Kea16]. This project consists of a number of scripts to pick up GSM traffic, and is written to work on a number of different SDRs. Among them USRP devices. To be more specific, lightly tweaked code from the programs *grgsm_scanner* and *grgsm_livemon* are used to identify active frequencies, and picking up traffic.

The first step in the traffic gathering process is to find which frequencies are being used by nearby BTS tower. To do this, lightly tweaked source code from the program *grgsm_scanner* is used. This program scans all possible GSM frequencies to see if they are active. It also get signal strength, and tries to get basic information about the cell, though it may fail at the latter activity. The python source code was changed so that it could be imported into other projects, superfluous terminal output was removed, and upon completion each active frequency and its signal strength is returned. *grgsm_scanner* is not perfect, and may return frequencies that are in fact not active.

To capture GSM traffic, the program *grgsm_livemon* together with the python library pyshark[New16] is used. *grgsm_livemon* utilizes the SDR to decode GSM traffic from a single frequency, which it then encapsulates with an ethernet frame, and sends to the loopback interfrace, also known as "lo". The only changes done to the source code is to remove a Graphical User Interface (GUI), as it is not used. The pyshark library is a wrapper for tshark, which lets python capture and parse packets.

For each frequency found, the tweaked *grgsm_livemon* is run as a subprocess. After which a pyshark capture object is created, and set to sniff the loopback interface. After the capture object has run for its allotted time, it is stored for further parsing. Pseudo code for the packet capturing process can be seen in algorithm 4.2.

Algorithm 4.2 Detect active GSM frequencies and gather broadcast traffic in pseudo code.

```
 // scan_speed comes from user input
frequencies = grgsm_scanner_tweak.full_frequency_scan(scan_speed);
for each frequency  in frequencies{
    process = subprocess.Popen(['grgsm_livemon', '-f', frequency ]);
    capture = pyshark.LiveCapture(interface='lo');
    capture.sniff(timeout=15+(5-scan_speed));
    os.kill(process.pid);
    frequencies[gsm_frequency].add_captured_traffic(capture);
}
```

Since *grgsm_scanner* will at times return inactive frequencies, the capture objects will not always contain any packets. If this is the case they are discarded.

When a new cell, or frequency belonging to that cell is detected it is stored together with the following information.

- CI
- MCC
- MNC
- LAC
- The set of frequencies, used by the cell.
- CRH
- CRO
- TO
- PT
- T3212
- RX_LEV_ACCESS_MIN
- MS_TXPWR_MAX_CCH

These values either identify the cell, or effect the cell selection and reselection process, as described in section 2.7. Every time a frequency is detected during a scan it is also stored, together with the location and signal power.

4.1.3 Tests

After each frequency and traffic scan the IMSI catcher detector runs a number of tests. Many of the tests described in section 3.2 are implemented, however

After all these steps are finished, any new information is added to a log file, which can be used at later scans or a separate analysis.

Chapter 5 describes an experiment where the IMSI catcher detector just described is tested in a live environment. Based on the results from this experiments the p practical realities of the tests mentioned above are discuses. Among them what threshold values a suitable, and further expansions to the set of tests.

Table 4.1: All events that will cause an alarm to be triggered.

	Description
1	If the MCC is known, and a cell with different MCC is detected.
2	If there are any frequencies known to not be in used, and one of those are detected.
3	If there are any frequencies known to not be in used, and one of those matches the reported cell frequency list of a cell.
4	If there are any frequencies known to not be in used, and one of those matches a frequency neighbor list(BA list) of a cell.
5	If none of the frequencies in a neighbor list are are active.
6	If a neighbor list is empty or only contains one frequency.
7	If a neighbor list is empty or only contains one frequency.
8	If a utilized frequency is not in its own neighbor list.
9	If a utilized frequency is not among its cell's frequency list.
10	If the periodic update timer T3212 timer is below a threshold value.
11	If the CRH value is above a threshold value.
12	If the optional CRO value is above a threshold value.
13	If the optional CRO value is not set.
14	If the optional PT value is below a threshold value.
15	If the optional PT value is not set.
16	If any value, transmitted from the same cell, changes between scans.
17	There is overlap between BA neighbor lists from two different operators(different MNC).
18	A frequency with a previously strong signal disappears.
19	Two frequencies with same CI have different cell frequency list.

4.2 Automatic Configuration of an IMSI Catcher

The second piece of software automatically configures an IMSI catcher. It gathers information about about nearby GSM towers, and uses this information to make decisions about how to most efficiently gather IMSIs. After a configuration is decided upon, it configures and runs OpenBTS[Net16] as an IMSI catcher. The program could easily be rewritten to configure another program with the same, or more, capabilities as OpenBTS, however OpenBTS has the advantage of being easily available, and can run on the same USRP hardware as the configuration script.

During development, what seems to have been a crash between different versions of USRP Hardware Driver (UHD), the open source hardware driver that controls the USRP device, of OpenBTS and the configuration script was discovered. Though it might be possible to fix this miss match, it was more time efficient side step this issue by running the configuration script on a separate Virtual Machine (VM) from OpenBTS. The system is therefore divided into two python scripts: *configure_bts_client.py* and *liveshark.py*. The first runs on the same machine as OpenBTS and runs commands to configure it, while the latter runs in the other VM and does the frequency scan, decides on a configuration, and sends it back to the first.

4.2.1 OpenBTS

OpenBTS is an open source GSM BTS developed by Range Networks[Net16]. It was the first implementation of the Um interface available for free, and replaces the somewhat complicated network of entities in GSM(see section 2.2). The goal of the project, as stated in [Ied15], was to simplify and reduce the cost of deploying GSM. With the developing world, and hard to reach areas in mind. Beside the free software, one only needs a relatively cheap USRP device, like the Ettus N200[Res16b] which costs about 1500$, and a laptop to run the software, to have GSM service. However, this potent mix of simple deployment and low cost makes it a common choice for those who wish to create or study IMSI catchers. Shown, for example, in [Pag10]. Alongside OpenBTS, Asterisk, SIP Message Queue (SMqueue), and SIP Authorization Server (SIPAuthServe) are respectively responsible for call switching, SMS handling, and authentication[Ied15].

Detailed instructions abut how to install OpenBTS can be found in in [Net16]

4.2.2 Configuration

In this particular configuration, the main goal of the IMSI catcher is merely to catch IMSIs effectively. No particular care is taken to target a specific group or person, it is indiscriminate. There is for example no care given to which MNC should be used. However, it has a secondary but more immediate purpose. The program will serve to

set up an IMSI catcher in the experiment described in chapter 5. Therefore, it serves also to help test the IMSI catcher detector described previously in this chapter. To do this, the configuration shows sign of several IMSI catching strategies that may not be overlapping. For example: When a location update procedure is induced by having a different LAC than nearby cells, you do not need to have a low T3212 to induce the same thing.

The first step is gathering broadcast traffic from nearby BTSs, and is identical to the process described in section 4.1.2. With information gathered, basis for further decisions is taken from the cell with the strongest signal, as it is a likely candidate for a nearby MS to camp on. The IMSI catcher copies its MCC and MNC directly from this cell.

In the stored list cell selection procedure the MS uses the BA list of the cell in which it is currently camping as a basis for its choice cell. When the choosing which frequency to use, the script takes advantage of this by choosing a frequency from the list which did not get picked up by the frequency scan.

The CI is set to be close to that of the strongest signal, so as to seem related to that one, but not equal to any CI detected.

The LAC is set to be close, but not equal, to that of the strongest signal, so as to cause a Location Update Request (LUR). Thus, an IMSI can be intercepted.

The T3212 timer is set to its minimum value of 1, which equals 6 minutes, and CRH is set to the maximum of 7.

RX_LEV_ACCESS_MIN and MS_TXPWR_MAX_CCH are set to the same values as those used by the strongest signal.

After all values have been decided upon, OpenBTS is started and configured through its command-line interface, OpenBTSCLI. From python, these instructions are called through the *subprocess* library's *call* function. Beside the values chosen, the auto configuration script makes sure OpenBTS transmits on full power, and that it will accept any MS that tries to connect.

After setup, any IMSI picked up can be seen using *tmsis* command in OpenBTSCLI.

```
$ cd /OpenBTS
$ sudo ./OpenBTSCLI
OpenBTS> tmsis
```

Algorithm 4.3 Configure OpenBTS in pseudo code.

```
for config_command in config_commands{
    subprocess.call(['sudo','OpenBTSCLI','-c', config_command])
}
subprocess.call(['sudo','OpenBTSCLI','-c', 'power 0'])

subprocess.call(['sudo','OpenBTSCLI','-c',
    'config OpenRegistration .*'])
```

4.2.3 OpenBTS limitations

The observant reader may have noticed that the CRO, PT or TO offset were not configured in the previous section. This is one of the limitations to using OpenBTS as an IMSI catcher. After all, OpenBTS was not designed to be a malicious tool, it was meant to be a simple and cost effective way of deploying GSM[Ied15]. As such, it is designed to offer cellular network service, not to attain IMSIs. That is merely a by-product. Unless upcoming versions implement it, you can for example not configure the BA list broadcasted by the USRP device at will. Thus disrupting the stored list cell selection procedure.

Some optional parts of the GSM standard have simply not been implemented. This is the case for CRO, PT and TO. Which are an optional part of the SI 3 reset octets[3GP05]. With a high CRO, one can increase the likelihood a MS will connect to the BTS, as described in section 2.7. This would obviously makes OpenBTS a more efficient at catching IMSIs. However, if one reads [Bur09] it becomes clear that the creators are well aware that OpenBTS can be used maliciously. It may be that they intentionally did not implement these features in the hope that OpenBTS would not be associated with such malicious activity.

Chapter 5
Testing the IMSI Catcher Detector

This chapter describes an experiment which tests the IMSI catcher detector described in chapter 4. It was set to scan GSM traffic at three separate locations in Trondheim, Norway. At one of these locations an IMSI catcher was introduced, running the OpenBTS software.

5.1 Equipment

- 2 x Ettus N200 SDR, with a SBX Rev. 5.1 daughter board, GPSDO kit and antenna.
- 4 x Ettus VERT900 vertical antennas
- 1 x Dell XPS 13 laptop(2015), Intel Core i5 2.2GHz-5200U CPU, 8 GB RAM, running the Ubuntu 14.04 Operating System (OS).
- 1 x Dell Latitude E6330 laptop, Intel Core i5-3320M 2.6GHz CPU, 8 GB RAM, running the Windows 7 OS.
- 2 x Biltema Litium ION batteries, 12V, 2.4 Ah
- 1 x Mascot 9061 12/6V converter DC/DC
- 1 x Thule backpack

5.2 Setup

Each of the two laptops ran the VirtualBox[Cor16] OS emulation software. The Latitude lapptop ran one instance of Ubuntu 14.04, while the XPS 13 ran two instances of Ubuntu 14.04. Each laptop was connected to one of the N200s by an ethernet cable.

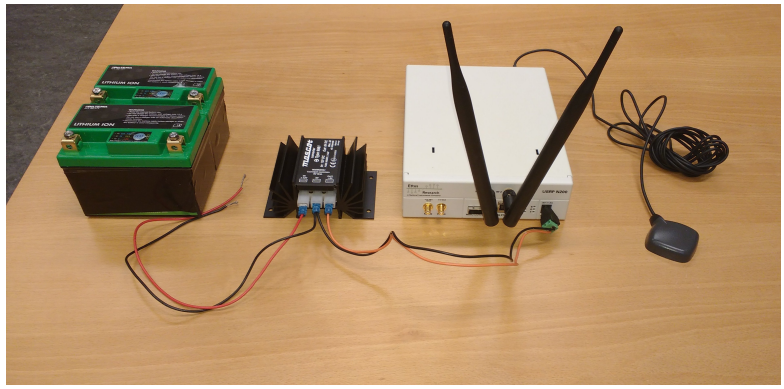

Figure 5.1: USRP setup. From left to right: 12V batteries(not connected), 12/6V converter, Ettus N200 with antennas, GPSDO antenna.

The two lithium ion batteries ran in parallel, and were connected to the 12/6V converter, which in turn was connected to the N200 connected to the latitude laptop. Thus powering it. The other N200 was connected to a power outlet. Each N200 had two VERT900 antennas connected to it.

For ease of transport the parallel batteries, converter and the attached N200 was placed in a backpack, while the Latitude laptop was carried by hand.

5.3 Procedure

At each location:

- Run the IMSI catcher detector script IMSI_Catcher_Catcher.py
- Set the MCC to 242(the MCC of Norway).
- Supply no frequencies that are known to not be in use.
- Attain GPS lock and position from GPSDO.
- Run scan with speed 3

At The Foundation for Scientific and Industrial Research (SINTEF) petroleum also run:

- Run the auto-configuration software liveshark.py and configure_bts_client.py to configure and start OpenBTS.

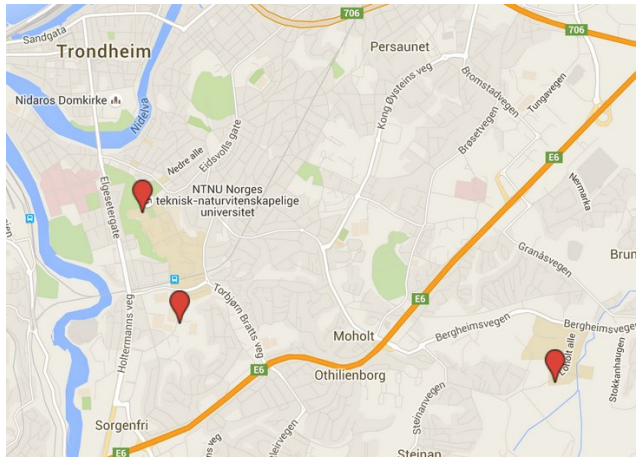

Figure 5.2: The three locations where scans were conducted. Created with Google Maps[Goo16].

- Run a RIMSI_Catcher_Catcher.py scan, retaining input from previous scans, at speed 3.
- Turn off OpenBTS.
- Run a IMSI_Catcher_Catcher.py scan at speed 3.

5.3.1 Locations

These are he three locations where scanned, as well as their GPS locations. They are also marked on Figure 5.2.

- Upper left: NTNU, campus Gløshaugen.
  GPS coordinates:
  $GPGGA,154107.00,6325.1294,N,01024.0419,E,1,09,1.3,67.6,M,40.0,M,,*55
- Lower left: SINTEF Petroleum parking lot.
  GPS coordinates:
  $GPGGA,074648.00,6324.6201,N,01024.4640,E,2,12,0.8,35.6,M,40.0,M,,*5E
- Lower right: NTNU, campus Dragvoll.
  GPS coordinates:
  $GPGGA,122203.00,6324.3691,N,01028.1940,E,2,12,0.8,158.7,M,39.8,M,,*64

Table 5.1: Overview of gsmlogg.xml.

Number of cells	41
Number of frequencies	41
MCC	242 - Norway's MCC
MNCs	01 - Telenor, 02 - Netcom
LACs	3305, 3331, 3304, 12411, 12431
Number of alarms triggered	40
Number of detected cells at Gløshaugen	19
Number of detected cells an Sintef Petroleum	27
Number of detected cells at Dragvoll	10
Power threshold(dBm)	-85
T3212	1
CRH threshold	7
CRO threshold	63
PT threshold	1

5.4 Data and Analysis

This section analyzes the data from the experiment described above. This data is in the form of a logg file called *gsmlogg.xml*, and is available at [Mru16]. It contains information on cells detected during scan, frequencies these cells used, parameters they broadcast, alarms triggered, and locations where the signals were picked up. Table 5.1 shows an overview of gsmlogg.xml. The threshold values are not in the logg file.

5.4.1 False positives

One test triggered more alarms than any other during the experiment. The 8th test mentioned in table 4.1: A frequency is not in its own BA neighborlist. It was triggered by 30 different cells from both MNCs, which accounts for 75% of all alarms triggered. This test obviously creates too many false negatives and should not be used.

The 18th test in table 4.1: A frequency with a previously strong signal disappears. Triggered an alarm 4 times. One of these was the IMSI catcher, which was turned off

Table 5.2: Average, Minimum and Maximum values per network operator.

	Telenor - 01	Netcom - 02
Num. base stations	18	23
CRH	Avg:3, Min:3, Max:3	Avg:2.2, Min:2, Max:7
CRO	Avg:0, Min:0, Max:0	Avg:0, Min:0, Max:0
TO	Avg:0, Min:0, Max:0	Avg:0, Min:0, Max:0
PT	Avg:20, Min:20, Max:20	Avg:20, Min:20, Max:20
T3212	Avg:40, Min:40, Max:40	Avg:38.3, Min:1, Max:40
RXLEV_ACCESS_MIN	Avg:-110, Min:-110, Max-110:	Avg:-110, Min:-110, Max:-110
MS_TXPWR_ MAX_CCH	Avg:5, Min:5, Max:5	Avg:5, Min:5, Max:5
Len. BA list	Avg:19.3, Min:15, Max:22	Avg:15, Min:1, Max:24
Len. cell's freq. list	Avg:3.05, Min:2, Max:7	Avg:1.95, Min:1, Max:2

as part of the procedure and expected to trigger an alarm. Even though it correctly triggered an alarm, there were three times as many false negatives. The experiment should be repeated with higher power threshold until a better value is found.

Excluding the alarms triggered by the IMSI catcher, the other 17 tests implemented in the IMSI catcher detector never trigger an alarm. While this does not say anything about their effectiveness at detecting IMSI catchers, it does mean that they do not trigger false negatives easily.

5.4.2 Parameter Differences Among Base Stations

This section examines all the BTSs detected as a whole. Table 5.2 shows the average, minimum, and maximum of each non-id parameter broadcast by a base station and the length of the BA list and cell frequency list.

Be aware that one of the Netcom BTSs is the IMSI catcher. It is this BTS responsible for the maximum CRH value, the minimum T3212 value, the minimum BA list length, and the minimum cell frequency list length. Since OpenBTS has not implemented CRO, PT, and TO, the IMSI catcher does not influence these values. Had the IMSI catcher values been removed there would be little difference between the BTSs in each operator. All cells use a low number of frequencies, though no

other cell uses less than 2. All Netcom cells use two frequencies, while the Telenor cells vary between 2 and 7. The length of the BA lists vary the most within each operator.

Because all parameter that impact the C1 and C2 values are the same within each operator, and CRO and TO are always 0, the C1 and C2 are identical. The only factor to impact selection and reselection among legitimate BTSs is the RLA. Coupled with the fact that CRO, and CRH are quite low compared their maximum values, an IMSI catcher could operate quite effectively in this environment. On the other hand, any deviance from the homogeneous network would stand out. Making any IMSI catcher that does not conform to the networks strict calibration easily detectable. The question of whether being detected is a risk or not, is up to the operator.

The IMSI catcher detector used in this experiment only keeps track of a small set of values broadcast by the BTSs. It should be expanded to encompass as many of the values as possible. After which the experiment should be repeated, and one could conclude whether or not the network traffic is uniform in other aspects as well.

This experiment was carried out over a small geographic area, and all within the same city, Trondheim. The experiment should be repeated at other locations to find whether such homogeneous networks are common, or Trondheim is the outlier.

5.4.3 The IMSI Catcher

Table 5.3 gives an overview of the IMSI catcher from the logg file. With seven alarms triggered, it was by far the cell with most alarms.

The following events caused an alarm to be triggered in by the IMSI catcher.

- Only one ARFCN in BA neighbor list.
- The ARFCN is not in the list of cell's list of ARFCNs.
- T3212 $= 1 \geq 1$, the threshold.
- CRH $= 7 \geq 7$, the threshold.
- CRO, though optional, is not set
- PT, though optional, is not set.
- TO, though optional, is not set.

Table 5.3: The IMSI catcher.

Parameter	Value
ARFCN	47
MCC	242 - Norway's MCC
MNC	02 - Netcom
LAC	3304
CRH	7
CRO	Not implemented
TO	Not implemented
PT	Not implemented
T3212	1
RXLEV_ACESS_MIN	-110
MS_TXPWR_MAX_CCH	5
BA list	47
Cell ARFCNs	0
Number of Alarms triggered	7

Let's consider the cause of each alarm triggered. There is only one ARFCN in the neighbor list because the OpenBTS can not be configured to advertise more ARFCNs. For the same reason, the ARFCN is not in the cell's ARFCN list. TheT3212 is 1, but did not need to be. The LAC has been changed to an otherwise nonexistent value. This would cause a location update procedure, and therefore an opportunity to catch the IMSI. The CRH is 7, the maximum. A useful value, if the goal is to mount a DOS attack. The CRO, PT, and TO trigger alarms because OpenBTS does not support them.

This means that 5 out of the 7 alarms were triggered because OpenBTS does not have support for them, and one alarm was triggered because the person conduction the experiment chose to do so, knowing it was not necessary. Leaving one alarm that could not be avoided by a specialized IMSI catcher and general knowledge about GSM. If the CRO, PT, and TO were implemented an attacker may have turned them up to increase the C1 value, and optimize the IMSI catcher. Though this would increase its visibility, as noted in section 5.4.2

5.4.4 Sources of Error

– The gr-gsm code base is a black box. Even though it is an open-source project, the author of this thesis has little to no experience with signal processing, and

is not qualified to assess it. A failing from its side may have influenced the outcome of the experiment.

– During the experiment, I observed that several frequencies were identified by the code from *grgsm_scanner*, but failed to retrieve enough, or any, traffic data. It is uncertain if they were actually non active frequencies, the allotted traffic gathering period is too short, or some other source of failure was to blame. Such frequencies are not added to the logg file. This should be amended, and the reason for failure be identified stored with with it.

Chapter 6
Conclusion

IMSI catchers and how they can be detected have been studied and presented in this thesis. First a background presentation of GSM was given. Then an overview of how IMSI catchers work and operated, followed by how they may be detected. A working IMSI catcher was created, and it was tested in a live environment.

The most surprising finding in the thesis is that network broadcast traffic in the Trondheim area of Norway, are very much alike. In fact in traffic originating from the cells with the same MNC, all parameters not connected with identification were alike. This indicates that trying to create a fingerprint of broadcast traffic could be a viable way of detecting IMSI catchers. However, this would require all network operators to configure their networks so that they are all very similar.

6.1 Further Work

The experiment is chapter 5 was only in a small geographical area. Gathering information should be expanded to more locations, to find out if nearby cells being configures to look similar is commonplace.

The IMSI catcher detector described in chapter 4 could be expanded to utilize several USRPs at once, by creating a network of USRPs connected to a centre node where calculations, tests and compared. This would require little work, but could not be done during the thesis, as more USRP devices were not available.

References

[3GP99] 3GPP. 3gpp t.s.03.12 location registration procedures v7.0. http://www.3gpp.org/DynaReport/43020.htm, aug. 1999. Accessed 2016.06.11.

[3GP05] 3GPP. 3gpp t.s. 05.08 radio subsystem link control v8.23. http://www.3gpp.org/DynaReport/0508.htm, nov. 2005. Accessed 2016.04.08.

[3GP09] 3GPP. 3gpp t.s. 44.018 mobile radio interface layer 3 radio resource control protocol. http://www.3gpp.org/DynaReport/44018.htm, jan. 2009. Accessed 2016.05.27.

[3GP14] 3GPP. 3gpp t.s. 23.003 numbering, addressing and identification v12.4.1. http://www.3gpp.org/DynaReport/23003.htm, oct. 2014. Accessed 2016.05.26.

[3GP16a] 3GPP. 3gpp t.s. 43.020 security related network functions v13.1. http://www.3gpp.org/DynaReport/43020.htm, apr. 2016. Accessed 2016.05.21.

[3GP16b] 3GPP. 3gpp t.s. 45.001 physical layer on the radio path. general description v13.1. http://www.3gpp.org/DynaReport/45001.htm, apr. 2016. Accessed 2016.05.21.

[3GP16c] 3GPP. 3gpp t.s. 45.002 multiplexing and multiple access on the radio path v13.1. http://www.3gpp.org/DynaReport/45002.htm, feb. 2016. Accessed 2016.05.21.

[3GP16d] 3GPP. 3gpp t.s. 45.005 radio transmission and reception v13.0.0. http://www.3gpp.org/DynaReport/45002.htm, feb. 2016. Accessed 2016.06.06.

[Aud08] Jan A. Audestad. *Technologies and Systems for Access and Transport Networks*. Artech House, 2008.

[bGG11] Supposedly by GAMMA Group. 3g-gsm tactical intercaption and target location. https://netzpolitik.org/wp-upload/Gamma-2011-GSM.pdf, 2011. Accessed 2016.06.01.

[Bur09] David Burges. 'openbts mailing list'. https://sourceforge.net/p/openbts/mailman/message/227914 2009. Accessed 2016.06.17.

[Cor16] Oracle Corporation. 'virtualbox home page'. http://www.virtualbox.org, 2016. Accessed 2016.05.20.

[Dab16] Adrian Dabrowski. 'technische universitat wien - internet security lecture notes 11: Mobile network security'. https://secenv.seclab.tuwien.ac.at/secenv/static/inetsec1/11_mobileNetworks.pdf, 2016. Accessed 2016.02.28.

[dPeT82] Conférence Européenne des Postes et Télécommunication. Gsm 2/82. http://www.gsm-history.org/19.html, june 1982. Accessed 2016.04.09.

[ea14] Adrian Dabrowski et al. Imsi catch me if you can: Imsi-catcher-catchers. https://www.sba-research.org/wp-content/uploads/publications/DabrowskiEtAl-IMSI-Catcher-Catcher-ACSAC2014.pdf, 2014. Accessed 2016.04.01.

[EFF13] EFF. As secretive "stingray" surveillance tool becomes more pervasive, questions over its illegality increase. https://www.eff.org/deeplinks/2013/02/secretive-stingray-surveillance-tool-becomes-more-pervasive-questions-over-its, 2013. Accessed 2016.01.18.

[Eve14] EventHelix. Gsm location update procedure. http://www.eventhelix.com/RealtimeMantra/Telecom/GSM_Location_Update_Sequence_Diagr 2014. Accessed 2016.06.12.

[FHMN10] Dan Forsberg, Günter Horn, Wolf-Dietrich Moeller, and Valtteri Niemi. *LTE Secuirty*. Wiley, 2010.

[Fou16] Python Software Foundation. https://www.python.org, 2016. Accessed 2016.06.01.

[Goo16] Google. 'google maps'. http://maps.google.com, 2016. Accessed 2016.05.26.

[Gro16] GAMMA Group. https://www.gammagroup.com/, 2016. Accessed 2016.06.01.

[22] GSM Association (GSMA). http://www.gsma.com/aboutus/gsm-technology/gsm, 2012. Accessed 2016.05.20.

[GSM09] GSMA. 'gsma statement on media reports relating to the breaking of gsm encryption'. http://www.gsma.com/newsroom/press-release/gsma-statement-on-media-reports-relating-to-the-breaking-of-gsm-encryption/, 2009. Accessed 2016.06.01.

[HH14] Kemal Huseinovice and Alisa Hebibovic. An approach to analyze security of gsm netowrk. https://www.researchgate.net/publication/269105419_An_Approach_to_Analyze_Security_of_ 2014. Accessed 2016.03.20.

[Hom] Third Generation Partnership Project Homepage. www.3gpp.org. Online. Accessed 2016.04.01.

[Ied15] Michael Iedema. *Getting Started with OpenBTS*. O'Reilly Media, 2015.

[Kea16] Piotr Krysik and et al. 'gr-gsm'. https://github.com/ptrkrysik/gr-gsm, 2016. Accessed 2016.04.02.

[Mru16] André Mruz. Thesis code. https://github.com/GreeneShoes/thesis, 2016.

[Net16] Range Networks. 'openbts'. http://www.openbts.org, 2016. Accessed 2016.05.26.

[New16] Kimi Newt(Pseudonum). 'pyshark'. https://github.com/KimiNewt/pyshark, 2016. Accessed 2016.03.27.

[NN03] Kaisa Nyberg and Valtteri Niemi. *UMTS Security*. Wiley, 2003.

[NP09] Karsten Nohl and Chris Paget. Gsm-srsly? https://events.ccc.de/congress/2009/Fahrplan/attachments/1519_26C3.Karsten.Nohl.GSM.pdf, 2009. Accessed 2016.06.17.

[OSM] OSMOCOMbb. http://bb.osmocom.org/trac/. Accessed 2016.06.01.

[Pag02] Paulo S. Pagliusi. A contemporary foreword on gsm security. In *in Proceedings Infrastructure Security International Conference (InfraSec 2002), LNCS 2437*, pages 129–144. Springer-Verlag, 2002.

[Pag10] Chris Paget. Practical cellphone spying (defcon 18). 'https://www.youtube.com/watch?v=DU8hg4FTm0g', 2010. Accessed 2016.06.01.

[Res16a] Ettus Research. https://www.ettus.com/product, 2016. Accessed 2016.06.01.

[Res16b] Ettus Research. 'ettus n200'. https://www.ettus.com/product/details/UN200-KIT, 2016. Accessed 2016.06.01.

[Sea15] Altaf Shaik and et al. 'practical attacks against privacy and availability in 4g/lte mobile communication systems'. https://arxiv.org/pdf/1510.07563v1.pdf, 2015. Accessed 2016.02.28.

[vdBVdR15] Fabian van den Broek, Roel Verdult, and Joeri de Ruiter. Defeating imsi catchers. http://www.cs.ru.nl/F̃.vandenBroek/Defeating_IMSI_Catchers.pdf, 2015. Accessed 2016.03.03.

[WM15] Susanne Wetzel and Ulrike Meyer. 'a man-in-the-middle attack on umts'. http://ece.wpi.edu/d̃chasaki/papers/mitm_umts.pdf, 2015. Accessed 2016.02.28.

Appendix A
Static IP

This appendix shows how one can set a static IP address on an interface in Ubuntu 14.04. In this example we want the interface eth1 to have the IP address 192.168.10.1. We also know that a network entity with IP address 192.168.10.2 will be eth1's gateway. Add the following lines to /etc/network/interfaces, but do not remove anything.

```
iface eth1 inet static
address 192.168.10.1
netmask 255.255.255.0
gateway 192.168.10.2
dns-nameservers 8.8.8.8
```

The Domain Name System (DNS) server 8.8.8.8 is a publicly available server hosted by Google, exchange with your own if needed. Remember to restart your machine after adding these lines.

To make sure the interface is up and running, run the following commands in a terminal.

```
$ sudo ifdown eth1
$ sudo ifup eth1
```

Appendix B
Configure Kernel Parameters

In order for software to run optimally when using USRP hardware, some kernel parameters may have to be changed. This appendix shows how to configure the OS Ubuntu 14.04 using the *sysctl* terminal command to do this.

```
$ sudo sysctl -w net.core.rmem_max=50000000
$ sudo sysctl -w net.core-wmem_max=1048576
$ sudo sysctl kernel.shmmni=32000
```

The first two commands change the User Datagram Protocol (UDP) read and write buffer sizes, respectively. The third command changes the system wide maximum number of shared memory segments

The values used in this example corresponds to those needed for the software described in chapter 4, running Ubuntu 14.04 and an Ettus N200[Res16b]. Other combinations of software and hardware may have other demands.

Appendix C
Installing GNU Radio

This guide shows how to install GNU Radio on a linux machine. Do the following commands in your terminal.

```
$ mkdir gnuradio
$ cd gnuradio
$ wget http://sbrac.org/files/build-gnuradio
$ chmod a+x ./build-gnuradio
$ sudo ./build-gnuradio -v
```

Make sure you have some spare time when doing this. The download and installation could take a long time.