NTNU
Norwegian University of
Science and Technology

Attacking Mobile Privacy By Catching MSISDN Location

Christoffer Evjen Ottesen

Master of Science in Communication Technology
Submission date:	June 2017
Supervisor:	Stig Frode Mjølsnes, IIK
Co-supervisor:	Ruxandra-Florentina Olimid, IIK

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Title:	Attacking Mobile Privacy By Catching MSISDN Location
Student:	Christoffer Evjen Ottesen

Problem description:

The MSISDN (or the phone number) is normally associated with a single mobile device linked to one person. Moreover, the phone number is used by many personal services and situations, such as registrations, communications, tickets and receipts, notifications, two-factor authentication, physical identification, geographical locations, and much more. Hence a mobile phone number can be considered to be personal information, and can be used to violate personal privacy.

Starting out with the idea and use of an IMSI Catcher, this master thesis work will investigate the possibilities of building an "MSISDN Catcher": a system that detects and links MSISDN with people and locations, without having direct access to the mobile device nor the mobile operator network and registries. However, the MSISDN Catcher can combine several information sources, including broadcasts and side channels that may be available, and try to exploit various technical and social attack mechanisms. A particular focus should be kept on the 4G mobile networks, where experimentation can be based on the OpenAirInterface open source software and USRP B200mini radio devices in our Wireless Security Lab.

Responsible professor:	Stig Frode Mjølsnes
Supervisor:	Ruxandra-Florentina Olimid

Abstract

Today, we find wireless technology almost everywhere. Easily, we can communicate with people on the other side of the world using our mobile phones. By June 2017, the global number of unique mobile subscribers is over 5 billion [29]. The fact that almost everyone carries a mobile device around combined with technological vulnerabilities, results in a delicate situation. Consequently, different parties with more or less legal intentions have performed attacks on the mobile technology. An IMSI Catcher is a popular device which utilises vulnerabilities in the networks.

The MSISDN also known as the phone number, is used by many personal services and situations. Hence a mobile phone number can be considered to be personal information, and can be used to violate personal privacy. Interestingly, there is not yet made MSISDN Catchers. Consequently, this master thesis investigates the possibility of building an MSISDN Catcher. With the main focus on the 4G LTE mobile network, the presence and security of the phone number in the architecture and relevant services were studied. Also, an open source IMSI Catcher was built based on the OpenAirInterface software and the USRP B200mini radio device. The IMSI Catcher was used in an experiment where the goal was to catch the IMEI identity of mobile phones camping near the base station.

It was found that there is no easy way to build an MSISDN Catcher based on the principle of an IMSI Catcher. Also, the open source IMSI Catcher was not able to catch the IMEI identity. Hence, these results show that both the identities are well protected against well-established exploit strategies.

Sammendrag

I dag finner vi trådløs teknologi nesten over alt. Med enkelhet, kan vi kommunisere med personer som befinner seg på andre siden av jordkloden ved hjelp av mobiltelefoner. I juni 2017, er det over 5 milliarder unike mobilabonnenter. Det faktum at nesten alle bærer på en mobiltelefon kombinert med teknologiske sikkerhetshull, resulterer i en kritisk situasjon. Som en konsekvens har ulike parter med mer eller mindre lovlige intensjoner utført angrep på denne mobilteknologien. En IMSI Catcher er en populær innretning som utnytter sårbarhet i nettverkene.

MSISDN også kjent som telefonnummer brukes i mange personlige tjenester og situasjoner. Derfor kan telefonnummeret kategoriseres som personlig informasjon og kan svekke personvernet til mobilbrukere. Det er interessant at det ikke er laget en MSISDN Catcher enda. Som et resultat av dette undersøkes muligheten for å bygge en MSISDN Catcher i denne masteroppgaven. Med hovedfokus på 4G mobilnettverk, ble lokasjonen og sikkerheten til MSISDN nummeret i nettverksarkitekturen og relevante tjenester studert. I tillegg ble det bygget en open source IMSI Catcher basert på OpenAirInterface programvare og en USRP B200mini. Den ble brukt i et eksperiment hvor målet var å fange IMEI identiteten til mobiltelefoner i nærheten av den improviserte basestasjonen.

Det ble avdekket at det ikke fantes noe enkel måte å bygge en MSISDN Catcher basert på prinsippene fra en IMSI Catcher. Dessuten klarte ikke IMSI Catcheren å fange IMEI identiteten fra telefoner i nærheten under eksperimentet. Derfor viser disse funnene at begge identitetene er godt beskyttet mot tradisjonelle angrepstaktikker.

Preface

This report is the result of my work during the Master period, which is the last semester of the 5-year Master of Science degree in Communication Technology at the Department of Information Security and Communication Technology at Norwegian University of Science and Technology (NTNU).

I would especially like to thank my responsible Professor Stig Frode Mjølsnes and Supervisor Ruxandra-Florentina Olimid for valuable discussions and guidance during the work of the pre-project as well as the Master thesis.

Also, I would like to thank all the people that have contributed to the OpenAirInterface project, Wireshark software, and the USRP B200mini.

Trondheim, 2017

Christoffer Evjen Ottesen

Contents

List of Figures ix

List of Tables xi

List of Algorithms xiii

List of Acronyms xvii

1 Introduction 1

1.1 Methodology 1

1.2 Outline 1

2 LTE 3

2.1 Architecture 3

2.1.1 The Evolved Packet Core (EPC) 4

2.1.2 MME 4

2.1.3 Serving Gateway 5

2.1.4 Packet Data Network Gateway 5

2.1.5 Home Subscriber Server 5

2.1.6 The access network - E-UTRAN 6

2.1.7 Circuit Switching (CS) 8

2.1.8 Packet Switching (PS) 8

2.2 Numbering, addressing and identification 9

2.2.1 International Mobile Subscriber Identity - IMSI 9

2.2.2 International Mobile Equipment Identity - IMEI 10

2.2.3 Mobile Station International Subscriber Directory Number - MSISDN 11

2.2.4 Globally Unique Temporary ID - GUTI 11

2.3 Numbering, addressing and identification within the IP multimedia core network subsystem 12

2.3.1 Home network domain name 12

2.3.2 Private User Identity 12

2.3.3 Public User Identity 13

2.4 LTE attach procedure 13
2.5 IP Multimedia Subsystem - IMS 20
2.5.1 Voice over LTE 24
2.5.2 VoLTE Architecture 25
2.6 VoLTE Security 26
2.6.1 IMS Registration 26

3 Tools 29
3.1 Open source IMSI Catcher 29
3.2 OpenAirInterface 30
3.2.1 OpenairCN - Core Network 30
3.2.2 Openair5G - Access Network 30
3.2.3 Hardware Requirements 30
3.2.4 Computer used for experiments 31
3.3 Wireshark 32
3.4 USRP B200mini 32

4 IMEI 35
4.1 Related Work 35
4.2 Experiment description 36
4.3 Configurations 37
4.3.1 Change requested identity type 37
4.3.2 Change definition of identity type 38
4.3.3 IMSI Catcher Setup 40
4.4 My Own Results 43
4.4.1 Sending modified identity request 43
4.4.2 Modified identity request with most significant bit = 1 44
4.5 Discussion 47

5 MSISDN Catchers 49
5.1 Overview 49
5.2 Related work 49
5.3 Security in LTE 50
5.3.1 The User Equipment 50
5.3.2 Field Test Feature on iOS 52
5.4 Security of the MSISDN in VoLTE 55
5.5 Leaks of private information in HTTP headers 55
5.6 Discussion 57

6 Conclusion 61
6.1 Further Work 62

References 63

Appendices

A OAI Installation 67

A.1 Kernel Configurations 67

A.2 Download OAI software 69

A.3 Installation of OAI EPC and eNB 69

B OAI eNB Machine Configuration 71

B.1 T3470 Timer 75

C USRP B200mini Series Architecture 77

List of Figures

2.1 EPS architecture using E-UTRAN access from [25] 4
2.2 Overview of the E-UTRAN architecture from [14] 6
2.3 Architecture for 3GPP access from [7] 7
2.4 Structure of IMSI from [10] 9
2.5 Structure of IMEI from [10] 10
2.6 Structure of MSISDN from [10] 11
2.7 Attach procedure [7] section 5.3.2 14
2.8 Architecture of LTE together with IMS 20
2.9 Transport of voice data from [37] 23
2.10 Transport of signalling data from [37] 23
2.11 VoLTE architecture from [13] 25
2.12 IMS registration process from [38] 26
2.13 VoLTE UE Attachment and IMS Registration message sequence from [13] 27

3.1 The B200mini board from [39] 33
3.2 The B200mini with enclosure kit and antennas 33

4.1 UE connecting to the network (IMSI requested) 43
4.2 Identity request message (IMSI) 44
4.3 Identity request message (IMEI) 45
4.4 UE connecting to the network (IMEI requested) 45
4.5 Identity request message (most significant bit = 1) 46
4.6 UE connecting to the network (most significant bit = 1) 46

5.1 EF MSISDN description from [4] 51
5.2 Screenshot of the Field Test main menu on an iPhone 6s 53
5.3 Screenshot of the SIM Info page showing the EF-MSISDN 53
5.4 Showing collected MSISDN by nationality. From [34] 56

A.1 i7z output from [20] 68

C.1 USRP B200mini Series Architecture from [39] 77

List of Tables

2.1 Use of switching types in GSM, GPRS, UMTS and LTE 8
2.2 Geographic regions identified by most significant bit in MCC 9

3.1 System specifications of desktop computer used in the IMEI experiment 31

4.1 Identity type structure in OpenAirInterface source code 38
4.2 TAC, MCC, and MNC values 41

5.1 MSISDN format 54

List of Algorithms

2.1 Derive Home Network Domain Name from IMSI 12
4.1 Relevant parts of the original source code 37
4.2 Changes in the code to allow 4-bit encoding 39
4.3 Setting most significant bit to 1 39

List of Acronyms

3GPP The 3rd Generation Partnership Project.

APN Access Point Name.

AS Application Servers.

AUC Authentication Center.

BCD Binary Coded Decimal.

BGCF Breakout Gateway Control Function.

CC Country Code.

CD Check Digit.

CLI Client Line Identification.

CMC Connection Mobility Control.

CN Core Network.

CoNC Cause of no CLI.

COTF Ciphered Options Transfer Flag.

COTS Commercial off-the-shelf.

CS Circuit Switched.

CSCF Call Session Control Function.

CSFB Circuit-Switched Fallback.

DOS Denial Of Service.

E-CSCF Emergency-CSCF.

EF Elementary File.

EIR Equipment Identity Register.

EMM EPS Mobility Management.

eNB Evolved NodeB.

EPC Evolved Packet Core.

EPS Evolved Packet System.

ESM EPS Session Management.

ETSI European Telecommunications Standards Institute.

E-UTRAN Evolved Universal Terrestrial Access Network.

FPGA Field Programmable Gate Array.

FQDN Fully Qualified Domain Name.

GPRS General Packet Radio Service.

GSM Global System for Mobile Communications.

GSMA GSM Association.

GUMMEI Globally Unique Mobility Management Entity Identifier.

GUTI Globally Unique Temporary ID.

GW Gateway.

HLR Home Location Register.

HSS Home Subscriber Server.

HTTP Hypertext Transfer Protocol.

HTTPS Hypertext Transfer Protocol Secure.

I-CSCF Interrogating-CSCF.

IMEI International Mobile Station Equipment Identity.

IMEISV International Mobile Station Equipment Identity Software Version.

IMS IP Multimedia Subsystem.

IMSI International Mobile Subscriber Identity.

IPSec IP Security.

ISIM IP Multimedia Services Identity Module.

ISP Internet Service Provider.

ITU-T International Telecommunication Union Telecommunication Standardization Sector.

iOS iPhone OS.

LRF Location Retrieval Function.

LTE Long Term Evolution.

MCC Mobile Country Code.

MGCF Media Gateway Control Function.

MGW Media Gateway.

MITM Man-In-The-Middle.

MME Mobile Management Entity.

MNC Mobile Network Code.

MNO Mobile Network Operator.

MRF Multimedia Resource Function.

MRFC MRF Controller.

MRFP MRF Processor.

MS Mobile Station.

MSIN Mobile Subscriber Identification Number.

MSISDN Mobile Station International Subscriber Directory Number.

NAI Network Access Identifier.

NAS Non Access Stratum.

NDC National Destination Code.

NMSI National Mobile Subscriber Identity.

NPI Numbering Plan Identification.

NTNU Norwegian University of Science and Technology.

OAI OpenAirInterface.

OS Operating System.

OSA OpenAirInterface Software Alliance.

PCC Policy and Charging Control.

PCIe Peripheral Component Interconnect Express.

PCO Protocol Configuration Options.

PCRF Policy and Charging Rules Function.

P-CSCF Proxy-CSCF.

PDN Packet Data Network.

PDN GW Packet Data Network Gateway.

PI Presentation Indicator.

PLMN Public Land Mobile Network.

PS Packet Switched.

PSTN Public Switched Telephone Network.

P-TMSI Packet-Temporary Mobile Subscriber Identity.

QOS Quality Of Service.

RAC Radio Admission Control.

RBC Radio Bearer Control.

RRC Radio Resource Control.

SCCP Signalling Connection Control Part.

S-CSCF Serving-CSCF.

SD Spare Digit.

Serving GW Serving Gateway.

SGSN Serving GPRS Support Node.

SI Screening Indicator.

SIB System Information Broadcast.

SIM Subscriber Identity Module.

SIP Session Initiation Protocol.

SMS Short Message Service.

SN Subscriber Number.

SNR Serial Number.

TAC Type Allocation Code.

TAI Tracking Area Identity.

TEID Tunnel Endpoint Identifier.

TLS Transport Layer Security.

TON Type of Number.

UE User Equipment.

UICC Universal Integrated Circuit Card.

UMTS Universal Mobile Telecommunications System.

URI Uniform Resource Identifier.

USIM Universal Subscriber Identity Module.

USRP Universal Software Radio Peripheral.

VoIP Voice Over IP.

VoLTE Voice over LTE.

Chapter 1 Introduction

1.1 Methodology

The methodology of the work done in this master thesis was first to become familiar with the specification documents of the Long Term Evolution (LTE) standard. The main source for the specification documents was The 3rd Generation Partnership Project (3GPP) website. Offering a good overview of the different specification subjects[1]. Also, relevant work published in academic papers was studied. In the literature study, the main focus was:

- Where is the telephone number stored in the architecture?
- What relevant procedures involve the telephone number?

Importantly, I early prioritised to become familiar with the open source LTE implementation by OpenAirInterface (OAI), the documentation found on their website[2], and installation/operation of the software. In that way, knowledge about the experimental possibilities was gathered early. Making it easier to think of possible relevant experiments as I evolved my understanding of the problem area.

1.2 Outline

Chapter 2 gives an introduction to the essential parts of the LTE standard related to the problem area of the thesis. Specifically, the overall architecture of the LTE Evolved Packet System (EPS) as well as the involved entities, are explained. Also, the IP Multimedia Subsystem (IMS) architecture is described with its respective entities as well as the Voice over LTE (VoLTE) technology and security. Addressing and Identification principles used in LTE and IMS are also covered.

[1] http://www.3gpp.org/specifications/specification-numbering

[2] http://www.openairinterface.org

Chapter 3 presents the tools used in the experimental part of the thesis. Including the open source LTE implementation by OAI, Universal Software Radio Peripheral (USRP), and system information about the desktop computer used in the experiment.

Chapter 4 presents the International Mobile Station Equipment Identity (IMEI) experiment. Including related work, experiment description, configurations, results, and discussion.

Chapter 5 presents the work done related to the possibility of catching the Mobile Station International Subscriber Directory Number (MSISDN) number. Specifically, the chapter looks into the security of the MSISDN storage locations and procedures involving the MSISDN. In the end, the findings are discussed.

Chapter 6 contains the overall conclusion of the thesis and a section about further work.

Chapter 2 LTE

Wireless mobile communication has developed through multiple generations. From 1G that was introduced in the 1980s where analogue radio signal where used. Later on, 2G was deployed in the 1990s introducing digital signals, data services like Short Message Service (SMS) and digital encryption of voice calls. Followed by 3G in 1998, with higher data rates and better security. Finally, 4G or LTE was introduced in specification 3GPP R8. The first commercial LTE network was deployed in Stockholm by TeliaSonera in 2009 [17]. Interestingly, LTE made up of the access network part Evolved Universal Terrestrial Access Network (E-UTRAN) and core network EPS are fully IP-based. The fact that both data and voice services are carried over packet switched networks allows higher transmission speeds compared to 3G where voice still was carried over circuit switched network. This chapter presents a technical overview of the LTE technology.

2.1 Architecture

This section will present the overall architectural design of the EPS. Interestingly, the EPS is fully based on IP using Packet Switched (PS) connections for all communication. Consequently, the network architecture has evolved from the solutions we find in the legacy mobile networks. Basically, the EPS is divided into two parts: the core network (Evolved Packet Core (EPC)) and the access network (E-UTRAN). The following subsections will present the overall architecture and types of switching in mobile networks. Including details about the entities found in both the EPS as well as E-UTRAN part.

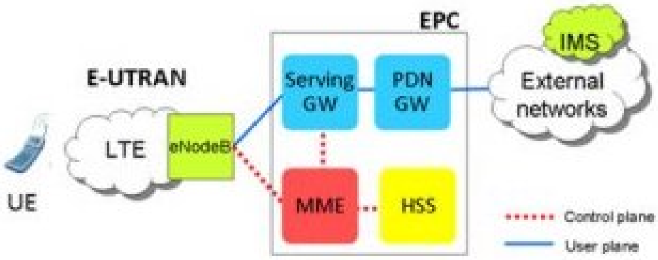


Figure 2.1: EPS architecture using E-UTRAN access from [25]

2.1.1 The Evolved Packet Core (EPC)

The LTE architecture is outlined in figure 2.3. Each of the elements in the architecture is going to be described further:

2.1.2 MME

The Mobile Management Entity (MME) can be seen as a control entity in the architecture. Briefly, the role of the MME is to support functions for the control plan in the EPS. Importantly, the MME is the termination point for the Non Access Stratum (NAS) and is responsible for handling mobility signalling as well as the security of the E-UTRAN access. Necessarily, the MME offers the listed functionalities below as described in the 3GPP specification document [9] section 4.1.4.1:

- NAS signalling and security
- Inter Core Network (CN) node signalling for mobility between 3GPP access networks
- Tracking Area List management
- Packet Data Network Gateway (PDN GW) and Serving Gateway (Serving GW) selection
- Serving GPRS Support Node (SGSN) selection for handovers to 2G or 3G 3GPP access networks
- Roaming

- Authentication
- Bearer management functions including dedicated bearer establishment
- Lawful Interception of signalling traffic

2.1.3 Serving Gateway

While the MME handles the control plane, the Serving GW and the PDN GW works with the user plane. Briefly, their tasks are to transport IP traffic between the User Equipment (UE) and external networks. As seen in figure 2.3, the Serving GW connects the radio-based side (E-UTRAN) and the EPC part. Moreover, for each UE there is a single Serving GW associated with EPS at a given point of time [46].

2.1.4 Packet Data Network Gateway

The PDN GW is where the EPC and the external IP networks interconnect. The PDN GW has responsibility for routing packets to and from the external Packet Data Networks. Additionally, the PDN GW allocates IP addresses and handles Policy and Charging Control (PCC). The PDN GW is logically connected to the Serving GW.

2.1.5 Home Subscriber Server

The Home Subscriber Server (HSS) entity contains subscriber-related and user-related information. Inside a home network, there can be multiple HSSs depending on the number of subscribers and how many subscribers each HSS can handle. According to the 3GPP specification [9], the HSS shall contain the following user related information:

- User identification, Numbering and addressing information
- User Security information: Network access control information for authentication and authorisation
- User Location information at inter-system level: the HSS supports the user registration, and stores inter-system location information
- User profile information

This information is used to provide functionality like authorisation, authentication, naming or address resolution, location dependencies and support to call control servers. Also, the HSS knows the identity of the current MME the user are connected to. In some cases, the Authentication Center (AUC) is integrated into the HSS. The AUC makes vectors for authentication and security keys. Additionally, the HSS contains subscription details for IMS services like VoLTE. Basically, the HSS supplies these subscription details to responsible entities during attach procedure and IMS registration. In the attach procedure, the subscription details are provided to the MME. Moreover, in the IMS registration procedure, subscription details are provided to the Serving-CSCF (S-CSCF).

2.1.6 The access network - E-UTRAN

E-UTRAN is the access technology used in LTE. The responsibility for the access network is to handle radio communicating to the connecting UEs. Basically, the access network is made up from base stations. In E-UTRAN, the base stations are named Evolved NodeB (eNB). Multiple eNBs can be directly connected to each other via the X2 interface. Further, the eNBs are connected to the core network (EPC) via the S1 interface. An overview of the access network architecture is shown in figure 2.2

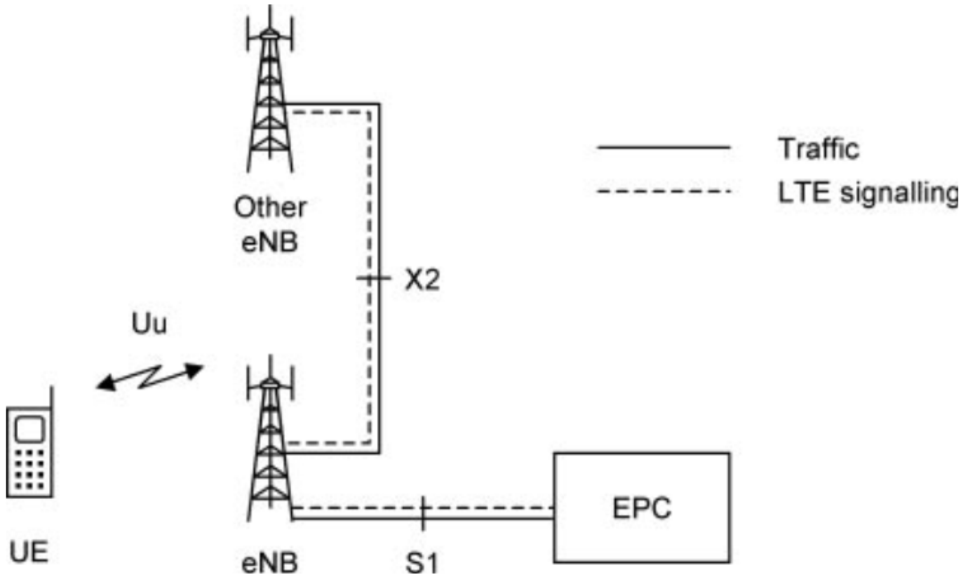


Figure 2.2: Overview of the E-UTRAN architecture from [14]

The access point - Evolved NodeB (eNB)

eNB is the type of base station used in E-UTRAN. It is responsible for providing the user plane and control plane terminations towards the UE [5]. In detail, according to [5], the eNB offers the following functionality:

- Functions for Radio Resource Management: Radio Bearer Control (RBC), Radio Admission Control (RAC), Connection Mobility Control (CMC), Dynamic allocation of resources to UEs in uplink, downlink, and sidelink
- IP header compression and encryption of user data stream
- Selection of an MME at UE attachment when no routing to an MME can be determined from the information provided by the UE
- Routing of User Plane data towards Serving GW
- Scheduling and transmission of paging messages (originated from the MME)
- Scheduling and transmission of broadcast information
- Measurement and measurement-reporting configuration for mobility and scheduling

The S1 interface connects the eNB to the EPC via the MME.

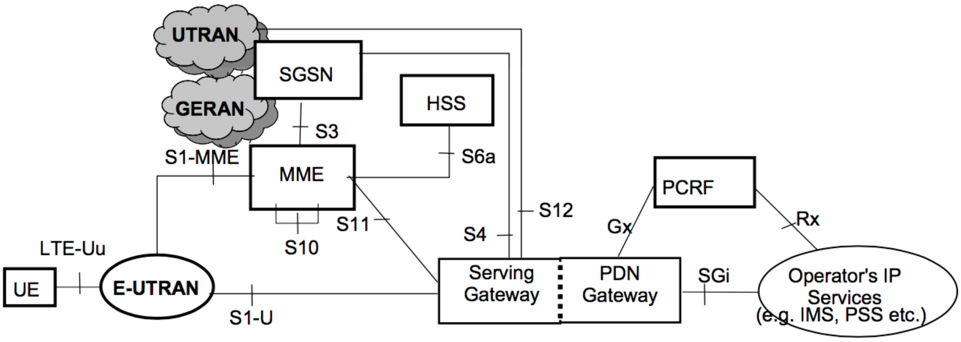


Figure 2.3: Architecture for 3GPP access from [7]

2.1.7 Circuit Switching (CS)

The definition of Circuit Switched (CS) connections can be explained by an example: Person A calls person B. In order to construct a connection between A and B, a solution is to assign an end-to-end 'circuit' dedicated for the communication between the two individuals over the mobile network. The circuit can be established across different parts of the network. These circuits can be seen as an evolution of the "two cans and a string" way of communication [25]. Once the circuit is established, the call can be established between A and B. The circuit remains active until the session is ended and eventually the circuit is destroyed. Interestingly, in the legacy network Global System for Mobile Communications (GSM), all communication (both voice and data) was sent over CS connections.

2.1.8 Packet Switching (PS)

In the PS technology, the need for establishment of dedicated communication circuits is removed. Basically, the solution is to transport data in packets. This technology was introduced with General Packet Radio Service (GPRS) specified by European Telecommunications Standards Institute (ETSI). Specifically, the PS technology was only deployed for data traffic in the beginning. Voice and SMS were still delivered over CS solutions in GPRS and Universal Mobile Telecommunications System (UMTS). Importantly, the release of LTE introduced a fully PS solution for both voice, and data traffic. Table 2.1 shows how the different switching types were used in the different mobile network generations.

	GSM	GPRS or UMTS	LTE
Circuit switched	Voice, SMS, Data	Voice, SMS	
Packet switched		Data	Data, Voice, SMS
	Only CS	CS and PS	Only PS

Table 2.1: Use of switching types in GSM, GPRS, UMTS and LTE

2.2 Numbering, addressing and identification

2.2.1 International Mobile Subscriber Identity - IMSI

The International Mobile Subscriber Identity (IMSI) number is a unique identification number allocated to each subscriber in the mobile network. IMSI is defined in Recommendation E.212 by International Telecommunication Union Telecommunication Standardization Sector (ITU-T) as a string of decimal digits, up to 15 digits, that identifies a unique mobile terminal or mobile subscriber internationally [30].

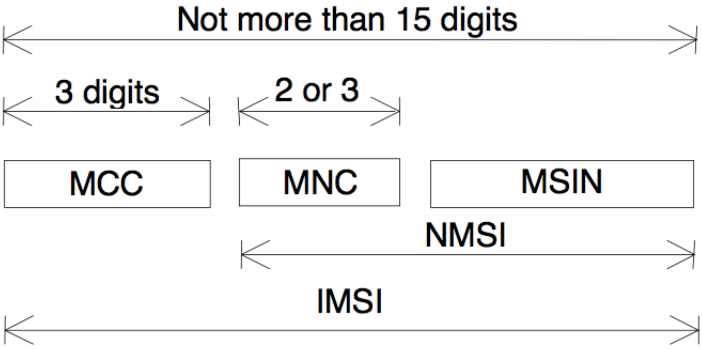


Figure 2.4: Structure of IMSI from [10]

Illustrated by figure 2.4, the IMSI consists of tree parts:

- Mobile Country Code (MCC) consisting of three digits. MCC is used to identify the country of domicile of the mobile subscriber. The most significant digit of the MCC identifies the geographical region.

Most significant digit	Geographical region
0	Test networks
2	Europe
3	North America and the Caribbean
4	Asia and the Middle East
5	Oceania
6	Africa
7	South and Central America
9	Worldwide

Table 2.2: Geographic regions identified by most significant bit in MCC

For instance, the MCC of Norway in Europe is 242 [31].

- Mobile Network Code (MNC) consisting of two or three digits. MNC identifies the home Public Land Mobile Network (PLMN) of the mobile subscriber. Where the usage of two or three digits depends on the value of the MNC [10]. The MCC combined with a MNC identifies a specific mobile network.
- Mobile Subscriber Identification Number (MSIN) consisting of 9 or 10 digits, identifies the mobile subscriber within a PLMN. MNC combined with the MSIN is known as National Mobile Subscriber Identity (NMSI) and is allocated by the national authority.

Conventionally, the allocation of IMSIs should be such that not more than the digits MCC + MNC of the IMSI have to be analysed in a foreign PLMN for information transfer [10].

2.2.2 International Mobile Equipment Identity - IMEI

The IMEI is a unique number for every mobile device using GSM, UMTS and LTE. It is usually found printed on the phone underneath the battery and can also be found by dialling the sequence *#06# [16]. Especially, the IMEI is used to validate UE's connecting to the mobile network. As a result, if the phone of a subscriber is stolen, the subscriber can report the IMEI number of the stolen phone to his or her network operator. Hence, the IMEI of the phone will be banned and it will not be able to connect to the mobile network. Making it useless, independent of what SIM-card the thief tries to use.

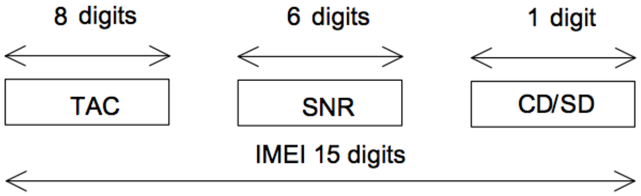


Figure 2.5: Structure of IMEI from [10]

Illustrated by 2.5, the IMEI consists of three elements:

- Type Allocation Code (TAC) consisting of 8 digits.
- Serial Number (SNR) consisting of 6 digits, is an individual serial number that uniquely identifies each UE within the TAC. Manufacturers shall allocate such individual serial numbers in sequential order [10].
- Check Digit (CD) / Spare Digit (SD).

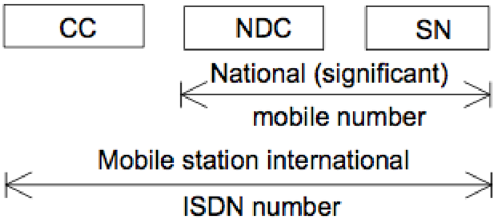

CC
NDC
SN
National (significant) mobile number
Mobile station international ISDN number

Figure 2.6: Structure of MSISDN from [10]

2.2.3 Mobile Station International Subscriber Directory Number - MSISDN

Illustrated by 2.6, the MSISDN consists of three elements:

- Country Code (CC) of the country in which the Mobile Station (MS) is registered
- National Destination Code (NDC)
- Subscriber Number (SN)

NDC + SN is known as National (significant) mobile number. Generally, a NDC is allocated to each PLMN. Furthermore, more than one NDC may be required for each PLMN in some countries [10]. Conventionally, the composition of MSISDN should be such that it can be used as a global title address in the Signalling Connection Control Part (SCCP) for routing messages to the Home Location Register (HLR) of the MS [10]. This is realised by traversing the information located in the CC and NDC of a particular MSISDN. Moreover, in case further routing is required, the first digits of the SN should hold the needed information.

2.2.4 Globally Unique Temporary ID - GUTI

The Globally Unique Temporary ID (GUTI) provides the network with a distinct identification type of the UE that does not reveal the UE or the permanent identity in the EPS. Additionally, it permits identification of the network and the MME. Specifically, the GUTI is composed of two main components:

- The first component uniquely identifies the MME that allocated the GUTI

- The second component uniquely identifies the UE within the MME that allocated the GUTI

2.3 Numbering, addressing and identification within the IP multimedia core network subsystem

2.3.1 Home network domain name

The form of the home network domain name is identical with the Internet domain name form. E.g. "example.com". The home network domain name can be derived from the IMSI number as defined by 3GPP [10]:

1. Take the first 5 or 6 digits (depending on the use of 2 or 3 digits in the MNC) and separate them into MCC and MNC. In scenarios, where the MNC is 2 digits, a zero shall be added at the beginning.
2. Now, use the MCC and MNC derived in the first step to create the "mnc<MNC>.mcc<MCC>.example.com" domain name.
3. Add the label "ims" to the beginning of the domain.

To demonstrate this, the home network domain name example from [10] is shown in 2.1:

Home network domain Name 2.1 Derive Home Network Domain Name from IMSI

```
IMSI in use: 234150999999999;

Then:

- MCC = 234
- MNC = 15
- MSIN = 0999999999

Which gives:
Home network domain name = ims.mnc015.mcc234.example.com
```

2.3.2 Private User Identity

Briefly, the Private User Identity shall take the form of a Network Access Identifier (NAI) resulting in the form username@realm. Specifically, the username corresponds to the IMSI number of the user, and the realm equals the home network domain name derived from the IMSI (As shown in 2.1). The private user identity is not used for the routing of Session Initiation Protocol (SIP) messages and is not present at all in IMS sessions (INVITE) and IMS session-unrelated procedures (Message, Subscribe, Notify) [38].

2.3.3 Public User Identity

The public user identity can be explained as the identity used by a user for requesting communication to another user inside the IMS subsystem. The form of the Public User Identity is SIP Uniform Resource Identifier (URI) or tel URI. When the telephone number or MSISDN is used as identity, the form is:

- **SIP URI**: sip:+<CC><NDC><SN>@example.com;user=phone
- **Tel URI**: tel:+<CC><NDC><SN>

Note that when the phone number or MSISDN is used as a private user identity, and the SIP URI form is used, a URI parameter (user=phone) must be added to the end, specify that the phone number is used.

2.4 LTE attach procedure

Necessarily, a user with an LTE compatible UE must register to the network in order to receive all services. The registration process is shown in detail in figure 2.7. This section will describe the E-UTRAN attach procedure in detail. Referring to the numbered messages in figure 2.7, each of them will be explained here. More detailed explanation of each message can be found in [7] in section 5.3.2.1

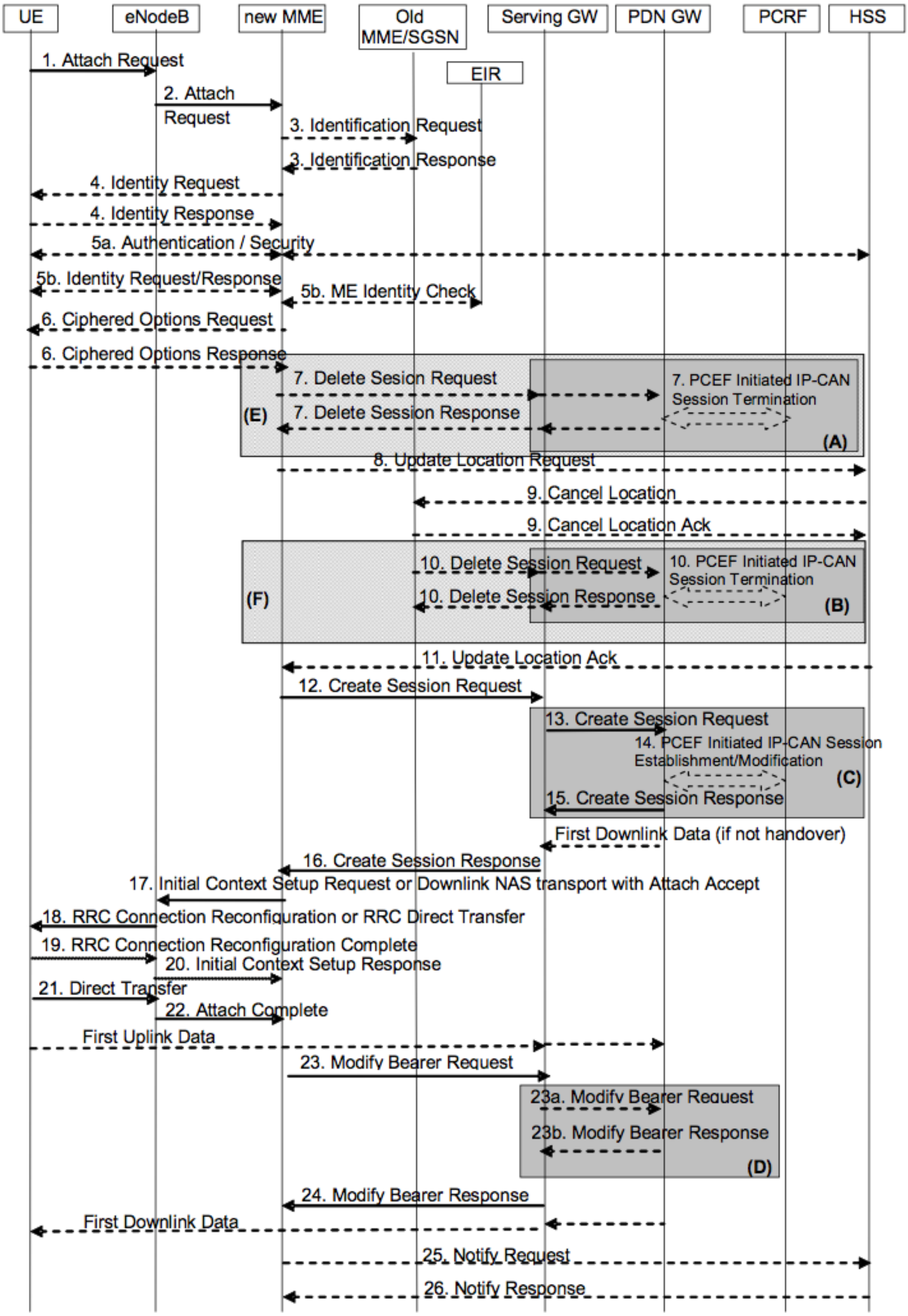

UE
eNodeB
new MME
Old MME/SGSN
Serving GW
PDN GW
PCRF
HSS
EIR
1. Attach Request
2. Attach Request
3. Identification Request
3. Identification Response
4. Identity Request
4. Identity Response
5a. Authentication / Security
5b. Identity Request/Response
5b. ME Identity Check
6. Ciphered Options Request
6. Ciphered Options Response
7. Delete Sesion Request
7. Delete Session Response
7. PCEF Initiated IP-CAN Session Termination
(E)
(A)
8. Update Location Request
9. Cancel Location
9. Cancel Location Ack
10. Delete Session Request
10. Delete Session Response
10. PCEF Initiated IP-CAN Session Termination
(F)
(B)
11. Update Location Ack
12. Create Session Request
13. Create Session Request
14. PCEF Initiated IP-CAN Session Establishment/Modification
(C)
15. Create Session Response
First Downlink Data (if not handover)
16. Create Session Response
17. Initial Context Setup Request or Downlink NAS transport with Attach Accept
18. RRC Connection Reconfiguration or RRC Direct Transfer
19. RRC Connection Reconfiguration Complete
20. Initial Context Setup Response
21. Direct Transfer
22. Attach Complete
First Uplink Data
23. Modify Bearer Request
23a. Modify Bearer Request
23b. Modify Bearer Response
(D)
24. Modify Bearer Response
First Downlink Data
25. Notify Request
26. Notify Response

Figure 2.7: Attach procedure [7] section 5.3.2

1. **Attach request**. The UE camping on an E-UTRAN cell reads the System Information Broadcast (SIB). Then, if the UE is allowed to attach, the next step is to send an Attach Request message to the eNB. The UE identifies itself to the network by including the IMSI or old GUTI. Additionally, Radio Resource Control (RRC) parameters to indicate selected network and the old Globally Unique Mobility Management Entity Identifier (GUMMEI) are included in the message. Also, if the UE holds last visited Tracking Area Identity (TAI), it is included in the Attach request message. Basically, these TAI values are useful for the MME in the process of building a list of TAIs from subsequent attach accept messages. Importantly, in the case of an emergency, the UE can initiate an Emergency Attach procedure. Briefly, this is done by setting the Attach Type and Request Type to "Emergency". Moreover, the UE shall send the valid GUTI or Packet-Temporary Mobile Subscriber Identity (P-TMSI). However, if the UE does not have a valid GUTI or P-TMSI, the IMSI or IMEI is included.

2. **Attach request**. Then, eNB reads the MME address from the RRC parameters that holds the old GUMMEI and the selected network, which the UE included in the Attach Request Message. Second, the eNB checks if the MME address is associated with the eNB. If there are no association, the new MME is selected by the MME selection function explained in [7] section 4.3.8.3. Finally, the attach request message is forwarded to the MME. In addition, the eNB informs the MME of the coverage level of the UE to help the location services [7].

3. **Identification request**. Firstly, in a scenario where the UE identified itself using GUTI and the MME has changed since last detachment, the new MME use the received GUTI to derive the old MME or SGSN address. Furthermore, the new MME sends an Identification Request to the old MME or SGSN to ask for the IMSI number. Depending on the type of the old node (MME or SGSN) the procedure is as follows:

   – The old node type is MME. Initially, the new MME sends an identification request (carrying the old GUTI + the complete attach request message) to the old MME asking for IMSI. Then the old MME verifies the attach request message by NAS MAC before responding with and identification response including the IMSI.

   – The old node type is SGSN.Just in the same way, the new MME sends an identification request (carrying the old GUTI + the complete attach request message). This time, the identification request is sent to the old SGSN. Then, the old SGSN verifies the attach request message by the P-TMSI signature. Lastly, the old SGSN responds with an identification request containing the MM context. Specifically, the MM context holds the IMSI number together with security related information [7].

In any event, if the old MME or SGSN do not recognise the UE or the verification fails, they respond with a suitable error message back to the new MME. In the situation of an emergency attachment, and the UE identifies itself with a temporary identity unknown to the MME, the MME shall instantly ask the UE for the IMSI. Whereas, if the UE identifies itself with IMEI, the MME will skip the IMSI request [7].

4. **Identity Request**. Whenever, the UE is not known by the old MME, old SGSN and new MME, an Identity Request is sent to the UE by the new MME. In that case, the UE shall answer with an Identity Response carrying the IMSI number. A timer named T3470 is used in case of a missing identity response. If an identity response is not received after the timer expires, the original identity request is re-transmitted to the UE and the timer is restarted. This process is repeated four times. Besides, the parameter "identity type 2" contained in the identity request message, specifies what type of identity the UE shall respond with.

5. **Authentication/Security and Identity Request/Response**

   – **5a** Granted that no UE context exists in the network or the Attach Request sent in (1) was not integrity protected, it is obligatory to activate integrity protection and NAS ciphering by authentication + NAS security setup. Authentication and NAS security setup could be skipped in case of an emergency attach. Therefore, from this point in the attach procedure, all the NAS messages are protected if not the UE is emergency attached.

   – **5b** In this step, the International Mobile Station Equipment Identity Software Version (IMEISV) is requested from the UE. With the exception of emergency attachment, the transmission of the IMEISV shall be encrypted. With the condition that a UE connects in emergency mode and identifies itself with the IMEI number, this step can be skipped. Moreover, to check if the connecting equipment is stolen and blacklisted, the MME may send the retrieved IMEI + IMSI to the Equipment Identity Register (EIR). The EIR is a database that has lists of blacklisted IMEI numbers (e.g. stolen equipment) and allowed IMEI numbers. After a search for the received IMEI in the register, the EIR responds back to the MME. Finally, the MME makes a choice to continue or reject the attaching UE based on the response received from EIR.

6. **Ciphered Options Request**. In the case where the UE has set the Ciphered Options Transfer Flag (COTF) in the initial Attach Request message in (1), the ciphered options shall be retrieved from the UE. These ciphered options could be Protocol Configuration Options (PCO), Access Point Name (APN) or both. Basically, they are used to handle scenarios where the UE has subscriptions

to multiple Packet Data Network (PDN)s. The UEs credentials and password contained in the PCO + the APN are then needed in the MME.

7. **Delete Session Request**. If the UE re-attaches to the same MME without completing the proper detach procedure, there can exist active bearer context in the new MME for the UE. Consequently, the new MME deletes these bearer contexts by sending Delete Session Request messages to the relevant Gateway (GW)s. Further, the new MME receives an acknowledgement from the GWs when the Delete Session Response messages are received. Finally, if Policy and Charging Rules Function (PCRF) is used, the PDN GW initiate an IP-CAN session termination procedure to indicate that resources have been released [7].

8. **Update Location Request**. An Update Location Request is sent to the HSS in these cases:

   – The MME has changed from last detachment
   – The MME has no valid subscription context for the connecting UE
   – The UE identify itself with its IMSI number in the Attach Request message
   – The old GUTI included in the Attach request sent by the UE has no valid context in the new MME
   – The PLMN-id of the TAI sent by the eNB is not coinciding with the value found in the GUTI in the UEs context.

9. **Cancel Location**. In this step, the HSS sends a Cancel Location message to the old MME. The Cancel Location message contains the IMSI of the UE + the cancellation type. Further, the old MME responds back to the HSS with a Cancel Location Ack message and tear down the old UE context.

10. **Delete Session Request**. Then, if the old MME or SGSN has active bearer contexts for the connecting UE, these bearer contexts are deleted by sending Delete Session Request messages to the GWs involved. Next, Delete Session Response messages are sent back from the involved GWs. Just in the same way as seen in step (7), if PCRF is deployed, an IP-CAN Session Termination procedure is executed.

11. **Update Location Ack**. After the old bearer contexts are deleted, the HSS now acknowledges the received Update Location Ack Message (from step 8) by sending an Update Location Ack message that carries the IMSI number and subscription data of the connecting UE. Importantly, this response message tells the new MME if the UE are allowed to attach to the network. If the subscription data of the UE violate the actual access restrictions (e.g. regional subscription restrictions), the UE can be denied access. Equally important, if the UE connects in emergency mode, the MME shall ignore a negative received Update Location Ack message and continue the attachment procedure.

12. **Create Session Request**. This step and step 12, 13, 14, 15 and 16 are only initiated in the case where an EPS Session Management (ESM) container was included in the attach request. Specifically, the ESM container includes these parameters; Request Type, PDN Type, PCO, COTF and Header Compression Configuration [7]. In the case of an emergency attachment, parameters from MME Emergency Configuration Data are applied for emergency bearer establishment [35].

13. **Create Session Request**. At this point, a new entry in the EPS Bearer table is made by the Serving GW. Further, the Serving GW sends a Create Session Request message to the PDN GW

14. **IP-CAN Session**. The PDN GW performs an IP-CAN Session Establishment procedure if PCC is deployed. Detailed information about this procedure can be found at [11].

15. **Create Session Response**. Before returning a Create Session Response message back to the Serving GW, the PDN GW creates a new entry in the EPS bearer context table and a Charging Id is generated.

16. **Create Session Response**. A Create Session Response message is sent from the Serving GW to the new MME.

17. **Initial Context Setup Request**. If an APN Restriction is received, the MME shall store the value for the Bearer Context [7]. Subsequently, the MME shall compare this received value with the Maximum APN Restriction value to ensure there is no conflict between the values.

18. **RRC Connection Reconfiguration**. In this step, the eNB sends the RRC Connection Reconfiguration Message including the EPS Radio Bearer Identity to the UE. Additionally, the Attach Accept message is sent to the UE.

19. **RRC Connection Reconfiguration Complete**. The UE, responds back to the eNB with the RRC Connection Reconfiguration Complete message.

20. **Initial Context Setup Response**. The eNB sends the Initial Context Setup Response message to the new MME. Included in this message are the eNBs corresponding Tunnel Endpoint Identifier (TEID) together with the eNB address used for downlink traffic.

21. **Direct Transfer** Next, a Direct Transfer message is sent from the UE to the eNB. The Attach Complete message is contained in the message.

22. **Attach Complete** Now, the eNB forwards the Attach Complete message received from the UE in the previous step, to the new MME.

23. **Modify Bearer Request** At this point, the new MME has received both the Initial Context Setup Response and the Attach Complete message from the UE. Subsequently, the new MME sends a Modify Bearer Request message to the Serving GW.

    - **23a**. In the situation where a handover indication is included in the Modify Bearer Request, a Modify Bearer Request indicating handover is sent to the PDN GW
    - **23b**. The PDN GW give a response in form of a Modify Bearer Response message sent to the Serving GW

24. **Modify Bearer Response** Further, the Serving GW acknowledges the received Modify Bearer Request from step 23, by sending a Modify Bearer Response message to the new MME.

25. **Notify Request** The new MME sends a Notify Request message to the HSS, holding the APN and PDN GW identity.

26. **Notify Response** Finally, the HSS saves the received identities and answer the MME with a Notify Response message.

2.5 IP Multimedia Subsystem - IMS

Since the EPS architecture in LTE uses only the PS domain for all communication types, some challenges can occur. Voice communication that has been carried over the CS domain in the legacy networks (see Table 2.1), shall now be transported in PS domain. An essential tool for solving this challenge is the IMS. The IMS can be seen as the collection off all CN elements for provision of multimedia services [8]. Moreover, the IP multimedia services utilise the IP access network and the multimedia transport capabilities to deliver traffic heavy content to the subscribers. Usefully, the IMS facilitates the original EPS network for offering multimedia services to the subscribers. Including, messaging, video, voice, data and web-based traffic. Figure 2.8 shows where the IMS is located in the overall EPS architecture.

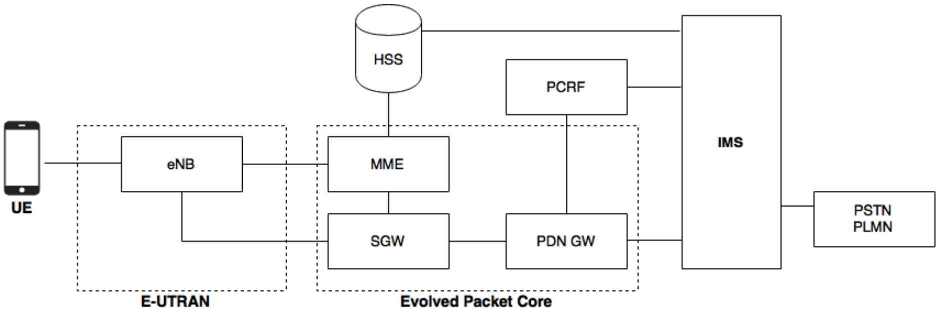


Figure 2.8: Architecture of LTE together with IMS

Basically, the IMS part connects to the PDN GW, PCRF, HSS and the Public Switched Telephone Network (PSTN) or PLMN. The responsibilities of these entities are as follows:

- The PDN GW is the EPC terminating point against the IMS part. Two bearers are established, running from the IMS through the PDN GW, the core network and out to the UE over the radio interface. Specifically, a signalling bearer + a dedicated bearer for transportation of media content such as data, video or voice are created.

- The PCRF is responsible for controlling the media flow. Briefly, typical tasks for the PCRF can be to make policy decisions for active subscribers in the network based on defined sets of rules or allocate needed bandwidth to dedicated call bearers used in VoLTE. This functionality, is valuable seen from the point of the Mobile Network Operator (MNO)s since the PCRF facilitates for differentiation of services. e.g. it is possible to charge subscriber a bigger amount in trade for more bandwidth or amount of data.

- The HSS holds detailed information related to identification and authentication of the subscribers. Similarly to the responsibilities the HSS has in EPC procedures, the HSS supports the IMS with access to the subscriber information database.

- PSTN or PLMN is the interconnection point.

The entities located in the IMS-part of Figure 2.8 are explained below. A more detailed overview of the IMS architecture can be seen in Figure 2.11.

- Application Servers (AS). Briefly, the responsibility of AS is to provide value-added services to the IMS network [37]. E.g. executing supplementary telephone services. Flexibly, the AS can be stationed within the home network or be offered from a third party. Moreover, the AS is involved in the processing of SIP messages.

- Call Session Control Function (CSCF) which is realised by:

  - Proxy-CSCF (P-CSCF). Seen from the UE side, the P-CSCF is the first interaction point with the IMS network. Basically, the functionality is similar to a proxy server. Requests from UEs or S-CSCF is transferred to S-CSCF or S-CSCF or UE. Other important tasks is establishment of IP Security (IPSec) association with UE during registration, resource availability checks, making information used to charge users and transfer emergency calls to Emergency-CSCF (E-CSCF).

  - S-CSCF offers session control services. In case of an incoming call, the S-CSCF forwards the received SIP INVITE message from Interrogating-CSCF (I-CSCF) to AS or P-CSCF depending on the service required. Further, the URI derived from the SIP INVITE message is replaced with the IP address of the called UE.

  - I-CSCF performs tasks on requests received from P-CSCF or S-CSCF. Specifically, when the first SIP REGISTER request is received, an S-CSCF is assigned to the UE and the message is forwarded there. This functionality also demands some exchanges of DIAMETER messages with the HSS [37]. When the second SIP REGISTER request and the first SIP INVITE message (incoming call), the I-CSCF sends a query to the HSS asking for the IP address of the S-CSCF associated with the UE before transferring the message to that S-CSCF.

  - E-CSCF is responsible for routing emergency calls forwarded from the P-CSCF to the emergency station closest to the UE the call originated from. This is done by utilising the Location Retrieval Function (LRF) which tells the location of the UE making the emergency call.

- Multimedia Resource Function (MRF) which involves:

  - MRF is responsible for controlling all media resources of the MRF Processor (MRFP). Also, it process information received from the S-CSCF and use this to facilitate the controlling of the MRFP.
  - MRFP is responsible for the generation of media flows within the regulations controlled by MRF Controller (MRFC). Moreover, it can work directly on the content of the media flows. e.g transcoding of audio media.

- The interconnection point towards PSTN or PLMN involves:

  - Breakout Gateway Control Function (BGCF). If an INVITE request is sent by the S-CSCF and the session cannot be forwarded to IMS network, the BGCF processes the INVITE message. This scenario is tied to calls made to subscribers connected to PSTN or PLMN. Also, the BGCF finds the next hop for routing of SIP messages by determining the Media Gateway Control Function (MGCF) responsible for internetworking with the PSTN or PLMN [37].
  - MGCF is responsible for connections in a Media Gateway (MGW). Including establishment, release and maintenance. The connection is basically, the association between the interface against PSTN or PLMN and the interface for the IP network.
  - MGW is the conversion station for the protocols of the multimedia flows between the endpoints (PSTN or PLMN and IP network). Figure 2.9 shows the relevant protocols between the end points. Also, eventual modifications done on the content of the media flow such as transcoding is located in the MGW as well as tones and announcements.
  - Serving GW is responsible for converting the signalling protocols found in the traffic between the MGCF and PSTN or PLMN. These signalling protocols are shown in figure 2.10

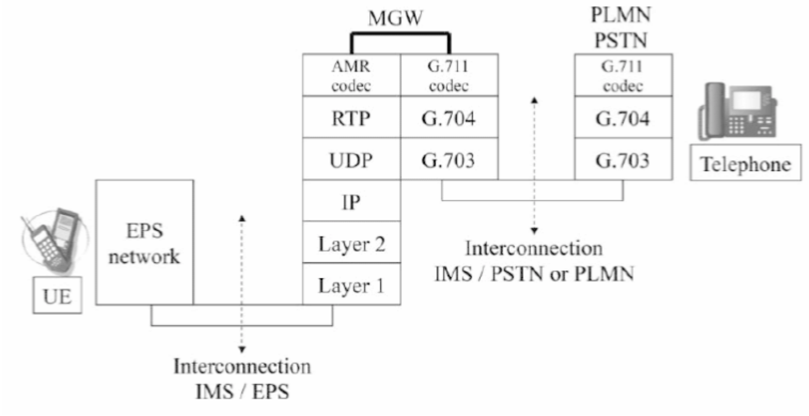


Figure 2.9: Transport of voice data from [37]

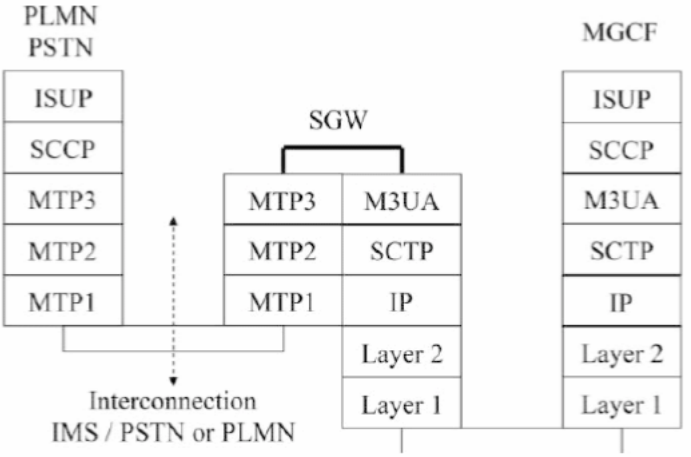


Figure 2.10: Transport of signalling data from [37]

2.5.1 Voice over LTE

Since the EPS is fully IP-based using only the PS domain (see figure 2.1), the MNOs implementing the LTE standard has two options for the previously CS approach for voice communication:

1. They can implement VoLTE, an IMS-based solution for carrying voice over a PS LTE network.
2. Circuit-Switched Fallback (CSFB). Briefly, CSFB is a solution for handling voice communication in LTE networks. Instead of using the VoLTE technology, CSFB offers voice communication to LTE connected subscribers by downgrading their connection to the legacy networks GSM or UMTS. In that way, utilising the existing CS architecture for handling voice communication.

VoLTE utilises the IMS technology for supporting voice traffic in the PS domain. The 3GPP developed many of the main components used in VoLTE a long time before it was implemented. The event that really started the collaboration towards deployment of VoLTE was when GSM Association (GSMA) announced the VoLTE initiative for driving the global mobile industry towards a standard way of delivering voice and messaging services for LTE on 15 February 2010 [1]. Leading MNOs and in total over 40 organisations from the mobile ecosystem backed the initiative. Already Marc 2010, the reference document on IMS profile for voice and SMS was published under the name IR.02 [28]. IR.92 is intended to ensure interoperable SIP-based IMS Voice Over IP (VoIP) and SMS for UE's and the LTE EPC [42]. Briefly, IR.92 specify the capabilities of the IMS technology. Additionally, supplementary services for telephony, transport, codecs, media negotiation, LTE radio, Quality Of Service (QOS) and bearer establishment is defined. Definitions of scenarios regarding roaming were defined in a separate document named IR.88 [27].

2.5.2 VoLTE Architecture

The VoLTE architecture is composed of:

1. VoLTE UE
2. Radio access network (E-UTRAN)
3. The LTE core network (EPC)
4. IMS core network. Responsible for providing the needed service layer for multimedia telephony. Specifically, the entities making up the IMS core are presented in Section 2.5.

Figure 2.11 shows a more detailed picture of the overall VoLTE architecture presented earlier in Figure 2.8. The respective entities inside the IMS domain, described in Section 2.5, can be seen in Figure 2.11 as well as the connections between them.

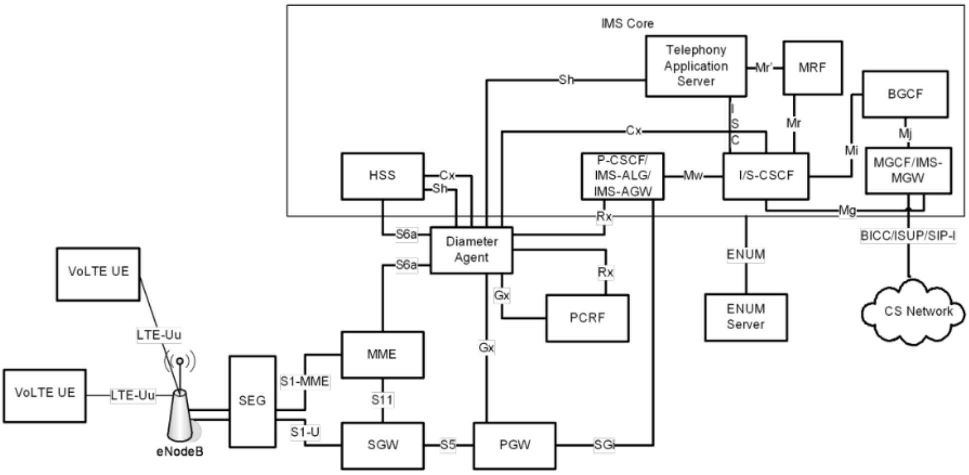


Figure 2.11: VoLTE architecture from [13]

2.6 VoLTE Security

Basically, if an MNO chooses to implement VoLTE for handling voice communication, it is important that attached subscribers are available for VoLTE services. Therefore, all UE's supporting VoLTE shall automatically perform an IMS registration after the LTE attach procedure [13]. The standard LTE attach procedure is explained in Section 2.4.

2.6.1 IMS Registration

The IMS registration process is initiated after the LTE attach procedure is completed. Figure 2.12 shows the message flow involved in the IMS registration process and the involved IMS entities. From a security perspective, the step initiating IPSec security associations between the UE and P-CSCF is important. From this point, SIP signalling involved in crucial procedures related to sessions, voice calls, SMS and more are guaranteed confidentiality as well as integrity. Consequently, VoLTE offers strong protection of traffic flowing over the wireless radio connection between UEs and the EPC.

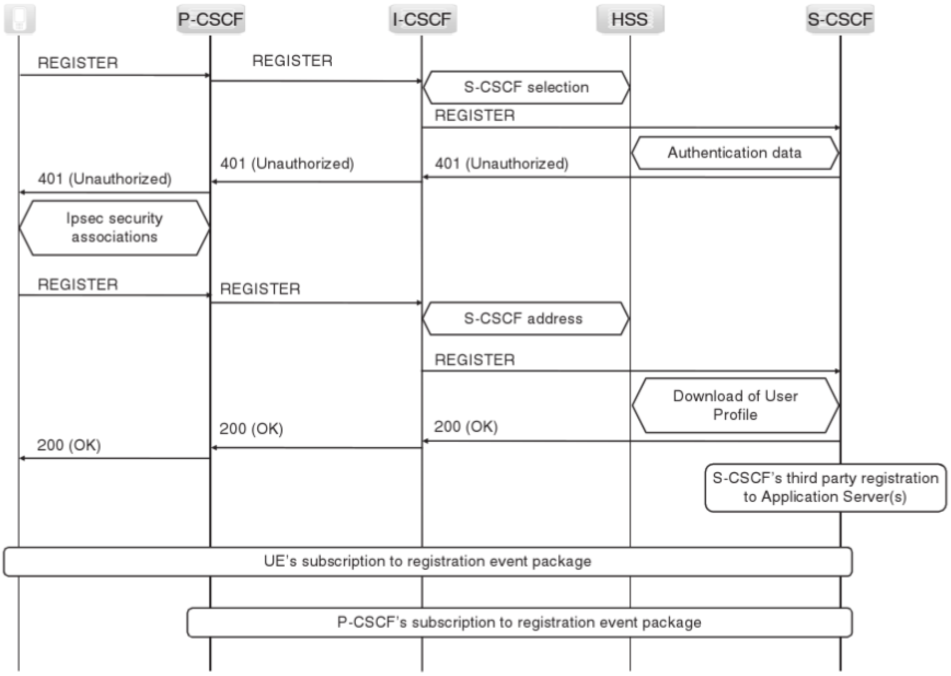


Figure 2.12: IMS registration process from [38]

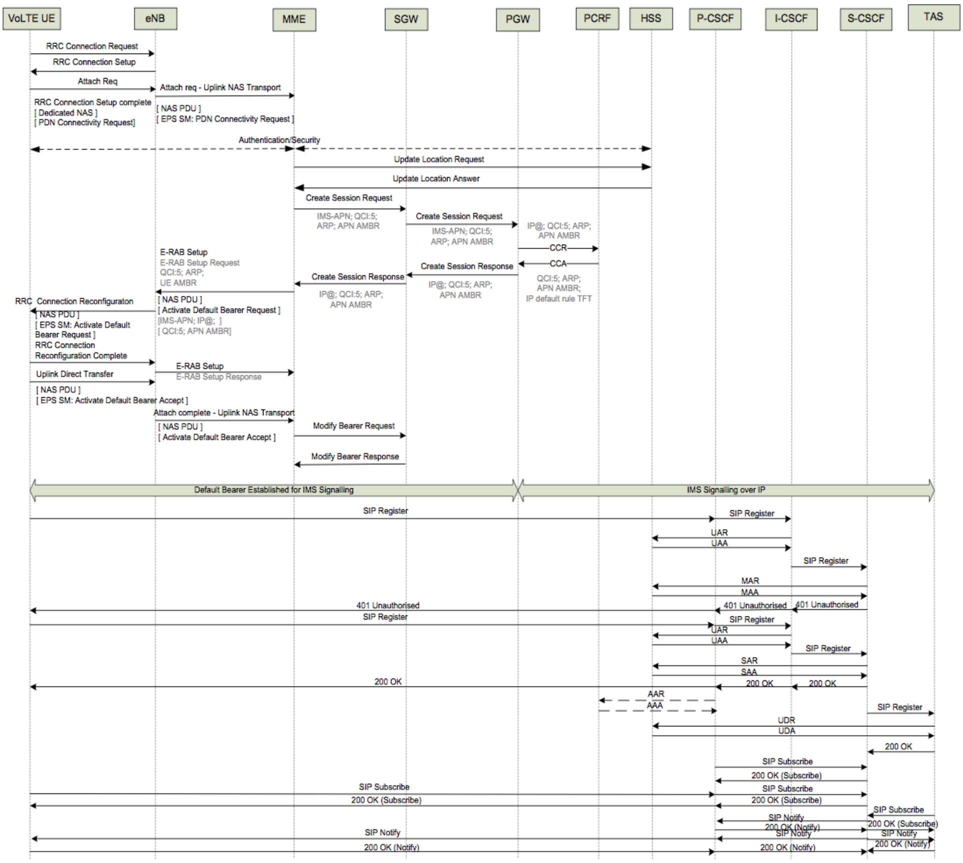


Figure 2.13: VoLTE UE Attachment and IMS Registration message sequence from [13]

Chapter 3 Tools

This chapter presents the tools used in this master thesis. Specifically, the principle of an open source IMSI Catcher is explained. Followed by a brief presentation of the OpenAirInterface software, the specifications of the computer used in the experiment, and information about the USRP B200mini radio device.

3.1 Open source IMSI Catcher

The principle of an IMSI Catcher is to launch Man-In-The-Middle (MITM) attacks on cellular mobile networks. The name originates from the first IMSI Catchers which was designed to reveal the IMSI number from the targets. Critically, such devices can be used to track handsets, obtain private subscriber information, intercept communication, launch Denial Of Service (DOS) attacks, and more. By acting as a base station and exploiting vulnerabilities in the mobile network, phones nearby are lured to drop their existing cellular connection and attach to the fake base station. Traditionally, hallmarks of the IMSI Catchers was expensive prices and limited mobility due to the high weight. Few manufacturers produced IMSI Catchers in the beginning and the economic barrier limited the device's use mostly to governmental agencies [15]. Despite this, smaller and cheaper devices were introduced on the marked during the last decade. Consequently, the former economic barrier for IMSI Catcher usage disappeared. At the same time, affordable USRP devices become available, making it possible to experiment with the technology at home. This stimulated growth in the open source community, resulting in open source mobile network implementations, compatible with public available hardware. Some of the open source projects are OpenBTS[1], OpenLTE[2], OpenAirInterface[3] and OpenBSC[4]. In the experiments in this thesis, the IMSI Catcher is based on the open source LTE

[1] http://openbts.org

[2] http://openlte.sourceforge.net

[3] http://www.openairinterface.org

[4] http://openbsc.osmocom.org/trac/

implementation by OAI, together with a USRP B200mini radio device. These tools will be explained next.

3.2 OpenAirInterface

The implementation of the LTE network used in this thesis is made by EUROCOM. More specifically, the software is made by OpenAirInterface Software Alliance (OSA). A separate entity from EUROCOM that specialises in providing an open source ecosystem for the EPC and E-UTRAN protocols of 3GPP cellular systems [22]. The open source model has been a huge success and aims to be a tool used by academia and the industry. Importantly, in the wireless industry, the major industrial professional's controls many of complex real-world systems. Therefore, the OSA is a great contributor for building a stronger relationship between the academia and the controlling industry part. Consequently, their knowledge and experiences can merge into a strong foundation that the 5th generation mobile network can be specified from.

The OAI LTE network implementation is divided into two parts; the core network (EPC) and the access network (E-UTRAN). In the source code, the core network part is named openairCN. On the other hand, the access network part is named openair5G.

3.2.1 OpenairCN - Core Network

OpenairCN implements the EPC part of the LTE standard. Including the respective entities: MME, HSS, Serving GW and PDN GW. Basically, the implementation is based on some parts of Release 10 LTE and is made for computers running Linux Operating System (OS).

3.2.2 Openair5G - Access Network

Openair5G implements the E-UTRAN functionality used in LTE, including the eNB implementation that can be launched on the USRP B200mini.

3.2.3 Hardware Requirements

This section will present the required hardware for running the OAI open source software. The full hardware requirement documentation is available at [21]. The OpenairCN (EPC implementation) shall work on all 64 bit Linux based computers. Challenges can arise when using container visualisation because the OpenairCN part demands kernel module installations. Additionally, for real-time operation of the software, the constraints are:

- USB 3.0 port is required for using hardware such as BladeRF, LimeSDR and Ettus USRPs together with the OAI software. Since a USRP is used in this thesis, the USB 3.0 port is essential on the computer dedicated to the experimentation.

- Ethernet transport requires a computer with an Ethernet port with speed of 10G or more.

- Usage of ExpressMIMO2 Peripheral Component Interconnect Express (PCIe) card made by EUROCOM requires a computer with an 8/16-way PCIe slot. To use a ExpressMIMO2 PCIe card together with a laptop running OAI software, a 1-way PCIe slot or ExpressCard slot can be used with an appropriate adapter.

Moreover, the OAI software requires computers based on Intel architecture for eNB or UE targets. Specifically, the software is well tested on the following processor families:

- Generation 3/4/5/6 Intel Core i5,i7

- Generation 2/3/4 Intel Xenon

- Intel Atom Rangeley, E38xx, x5-z8300

3.2.4 Computer used for experiments

For the IMEI experiment (Chapter 4), a Dell Optiplex 7040 desktop computer was used. The system specifications are shown in Table 3.1. Notably, the Intel CPU type is a member of Intel Generation 6 which is well tested for running the OAI software.

CPU	Intel (R) Core(TM) i7-6700 CPU @ 3.40GHz
Memory	32 GB
OS	Ubuntu 14.04 64-bit
kernel	3.19.0-61-lowlatency
Graphics	Intel@skylake DT GT2
Storage	500 GB HDD
USB version	3.0

Table 3.1: System specifications of desktop computer used in the IMEI experiment

3.3 Wireshark

In the mission of analysing the generated traffic from UEs connecting to the open source IMSI Catcher, Wireshark is used[5] in this master thesis. Wireshark is the world's foremost and widely-used network protocol analyser [45]. During the setup of the open source IMSI Catcher explained in Section 4.3.3 in Chapter 4, a tcpdump session is started. After completing an experiment, the tcpdump session was stopped, and the captured traffic is saved to a Packet Capture (PCAP)-file. Wireshark is used to open this PCAP-file, offering deep inspection of all the involved protocols.

3.4 USRP B200mini

The USRP B200mini is made by Ettus Research and is available from their website [39]. Especially, the price at 733 USD makes this a very affordable way of building an IMSI Catcher. The package includes the USRP B200mini board, a USB 3.0 cable and getting started guide. In addition to be highly affordable, the compact size of the board makes it very mobile. Technically, the B200mini board offers great flexibility for different tasks thanks to the programmable XilinX Spartan-6 Field Programmable Gate Array (FPGA). The frequency range for the board is 70 MHz to 6 GHz. Which means it can be used for all the relevant frequencies used in LTE [6]. Moreover, the B200mini board is powered by the USB 3.0 cable which is connected to the host computer. Figure C.1 in Appendix C shows the logical architecture of the board. Figure 3.1 shows the B200mini board. Moreover, Figure 3.2 shows the B200mini with the enclosure kit mounted as well as two antennas. The enclosure kit protects the board. With that said, the OAI software crashed multiple times during the IMEI experiment in Chapter 4. This was probably caused by overheating of the B200mini board after running the USRP over a longer time. Drilling many holes around the enclosure plate lowered the internal temperature enough to avoid the problem.

[5] https://www.wireshark.org

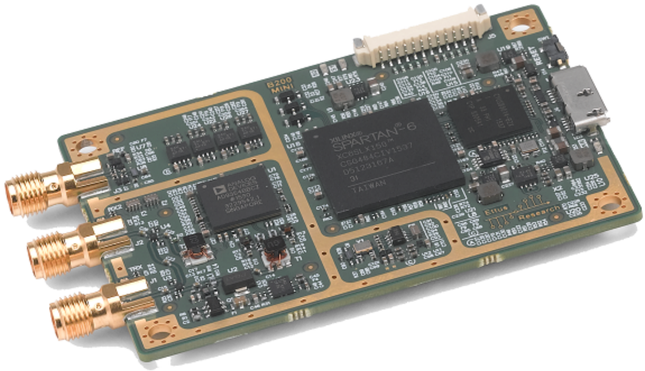


Figure 3.1: The B200mini board from [39]

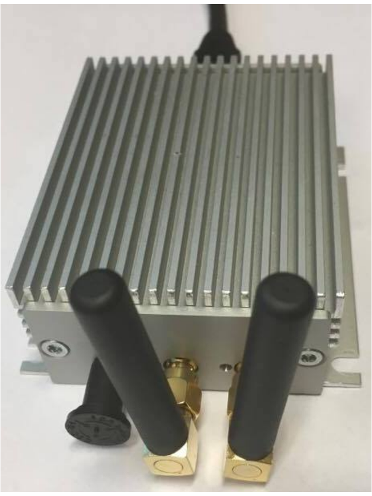


Figure 3.2: The B200mini with enclosure kit and antennas

Chapter 4 IMEI

In the process of building an MSISDN Catcher, the IMEI identity can be useful additional information. E.g. imagine a scenario where an IMSI Catcher modified to retrieve both IMEI and MSISDN identities is placed in an open area. The person operating the IMSI Catcher has two persons in sight. One of the persons holds a Samsung phone, the other person holds an iPhone. For simplicity, assume that only these two phones connect to the IMSI Catcher. Now, the IMSI Catcher provides the attacker with two identity pairs holding the MSISDN and the IMEI number. Then the attacker uses an online IMEI lookup service, which returns information about the brand or model of the mobile device[1]. The first searched IMEI number, turns out to belong to the Samsung phone. With that information, the attacker can link one of the caught telephone numbers to the person that holds the Samsung phone. Further, a lot of side channels can be used to find information tied to this specific telephone number. Motivated by this, an experiment with the goal of revealing the IMEI using an open source IMSI Catcher is presented in this chapter. A thorough explanation of the IMSI Catcher setup as well as all the changes in the source code are included. The experimentation was done in our Wireless Security Lab at times when there were few people on campus.

4.1 Related Work

In the paper 'How to not break LTE crypto' [33], they try to reveal the IMEI in clear. Requesting the IMEI in clear did not work in their case. With that said, they found that setting the most significant bit in the definition of the identity types did bypass the check in the modem of one of the handsets they used [33]. Their LTE testing infrastructure consisted of a standard eNB from Amarisoft, combined with a custom-made tiny LTE core network available at [32]. Moreover, in 2015, Torjus Retterstøl studied IMSI Catchers in his master thesis at NTNU, and he built and configured an IMSI Catcher based on USRP and OpenBTS [40].

[1] http://www.imei.info

4.2 Experiment description

In this experiment, an open source IMSI Catcher based on the OpenAirInterface software and USRP B200mini radio device is built. By default, the UE is asked to provide its IMSI for identification purposes during the LTE attach procedure. This is done when the network sends an identity request message to the UE (described in step 4 of the LTE attach procedure in Chapter 2). Specifically, the identity request procedure can also be used to ask for the IMEI identity. Starting out with the open source LTE implementation by OpenAirInterface, the identity request procedure will be located in the code. Further, this code will be manipulated so that the EPC asks the UE to provide the IMEI instead of the IMSI identity. Additionally, the successful method for revealing the IMEI found in the related work [33] will be tested.

4.3 Configurations

4.3.1 Change requested identity type

The identity request procedure is initiated by the core network and is therefore implemented in the Openair-CN part of the OAI source code. Moreover, the identity request procedure is part of the NAS that contains all non-radio signalling traffic between UE and MME. NAS contains the protocols EPS Mobility Management (EMM) and ESM. EMM holds the network-initiated identification procedure that is going to be located in the source code [24]. Looking into the source code of OAI, the EMM-folder is located in the file path "openair-cn/SRC/NAS/EMM". The parameter specifying the requested identity type is found in the file "Identification.C".

The code shown in 4.1, is the relevant part to specify what identity type the core network should ask the UE for. Normally, the requested identity type is IMSI. To change this to IMEI, the 'type' parameter on the last line in 4.1 must be changed from 'type' to '2'. Where '2' corresponds to the index where "IMEI" is located in the array named "*_emm_identity_type_str" shown at the beginning of 4.1. After this change is done, next time a mobile phone attaches to the IMSI Catcher, the identity request message will ask the UE to provide the IMEI.

Identification.c 4.1 Relevant parts of the original source code

```
/* String representation of the requested identity type */
static const char           *_emm_identity_type_str[] = {
"NOT AVAILABLE", "IMSI", "IMEI", "IMEISV", "TMSI" };

/*
 * Set the type of the requested identity
 */
data->type = type;
```

4.3.2 Change definition of identity type

After examining the OAI source code, it was found that the code needed some changes to be able to test the successful method for revealing the IMEI found in the related work [33]. Referring to the binary values representing the different identity types as shown in Table 4.1, it can be seen that the binary values are encoded using 3 bits. This is how it is done in the OAI source code. Which limits possible decimal values to 0-7. Specifically, in the source code used in the related work, the identity type definition is encoded using 4 bits, where only the 3 least significant bits are used to define the different identity types. In the successful attempt, the unused most significant bit (the fourth bit) was set to one. Resulting in the binary value 0b1010 for the IMEI (equal to 10 in decimal value). Hence, the encoding used in the original OAI source code (using 3 bit) does not allow to define the binary value of the IMEI identity equal to a decimal value of 10. It was found that just adding an extra bit in the identity definition of the OAI source code, and setting this bit equal to 1 did not work. Basically, the working solution is to change the encoding used to define the identity types from 3-bit encoding to 4-bit encoding in the OAI source code.

Identity type 2	Binary value	Decimal Value
IMSI	0b001	1
IMEI	0b010	2
IMEISV	0b011	3
TMSI	0b100	4

Table 4.1: Identity type structure in OpenAirInterface source code

In order to change the bit encoding of the identity types from 3-bit to 4-bit encoding, the following was done:

1. First, specify IMEI as the requested identity. This is done in the same way as explained in Section 4.3.1.
2. Modify the encoding of the identity type definition to use 4-bit instead of 3-bit. Firstly, the file where the identity type definition is coded must be found. Searching the CN part of the OAI source code, the definition was found in the file IdentityType2.c in the file path "openair-cn/SRC/NAS/IES/IdentityType2.c". The working changes in this file to allow 4-bit encoding are shown in 4.2
3. Now, the source code allows specifying 4 bits in the definition of the identity types. In the file "IdentityTYpe2.h" located in the file path "openair-cn/SRC/NAS/IES/IdentityType2.h" in the source code, the definition of the identity types can be changed on bit-level. Change the definition of the IMEI identity from "0b010" to "0b1010". The way to do this is shown in 4.3.

IdentityType2.c 4.2 Changes in the code to allow 4-bit encoding

```
Line 46:
change
"*identitytype2 = *buffer & 0x7;"
to
"*identitytype2 = *buffer & 0xF;"

Line 64:
change
"*identitytype2 = *buffer & 0x7;"
to
"*identitytype2 = *buffer & 0xF;"

Line 88:
change
"*(buffer + encoded) = 0x00 | (iei & 0xf0) | (*identitytype2 & 0x7);"
to
"*(buffer + encoded) = 0x00 | (iei & 0xf0) | (*identitytype2 & 0xF);"

Line 105:
change
"*(buffer + encoded) = 0x00 | (iei & 0xf0) | (*identitytype2 & 0x7);"
to
"*(buffer + encoded) = 0x00 | (iei & 0xf0) | (*identitytype2 & 0xF);"
```

IdentityType2.h 4.3 Setting most significant bit to 1

```
    Line 30:
    change "#define IDENTITY_TYPE_2_IMEI  0b010"
    to "#define IDENTITY_TYPE_2_IMEI  0b1010"
```

4.3.3 IMSI Catcher Setup

Before the experiment can be executed, the setup of the OAI EPC network, as well as the OAI eNB must be built and started with the right configurations. The process is as follows:

1. Complete the kernel configurations, source code download, and installation as explained in Appendix A.

2. Next, connect the USRP B200mini to the desktop computer with a USB 3.0 cable.

3. Further, the OAI eNB, EPC and HSS is built. To do this, automated build scripts included in the OAI source code need to be executed. This is done by running the code below in a terminal window:

   ```
   $ cd ~/openairinterface
   $ source oaienv
   $ cd cmake_targets
   $ ./build_oai -I  --eNB -x --install-system-files -w USRP
   $ ./build_oai -I  --eNB -x --install-system-files -w EXMIMO
   $ ./build_oai -I  --eNB -x --install-system-files -w BLADERF
   ```

   Note: Since the USRP B200mini is used in this experiment, only the first ./build command needs to be executed (for USRP). For explanation about the options used in the command, run ’./build_oai -h’ to show the explanation of each option. Importantly, the build process takes some time and will ask for a password that will be used to access the MySQL database as a root user. The HSS entity is preconfigured with ’linux’ as the password for accessing the MySQL database. Consequently, it is recommended to use ’linux’ as the password to simplify the configuration process.

   Similarly, to run and build the automated scripts for the openair-cn part of the OAI source code, run the following code in the terminal:

   ```
   $ cd openair-cn
   $ cd SCRIPTS
   $ ./build_mme -i
   $ ./build_hss -i
   $ ./build_spgw -i
   ```

   NOTE: These commands only needs to be executed one time for installing the missing packages [19].

4. Configuration: The needed kernel configurations, download and installation of OAI EPC and eNB, are presented in Appendix A. Configurations of the OAI eNB machine is explained in Appendix B. Including how to change MCC, MNC, and TAC. Table 4.2 shows the values used in this experimental setup.

Parameters	Value
Tracking Area Code (TAC)	1 (Incremented)
Mobile Country Code (MCC)	242
Mobile Network Code (MNC)	06

Table 4.2: TAC, MCC, and MNC values

The value of TAC was incremented by one each time the eNB was restarted. This was done because the tracking area update request message is not sent from phones near the base station unless the observed TAC of the eNB has changed.

By doing so, it is not necessary to restart the mobile phones between each attempt. The MCC of Norway (242) was used [41].

5. Running eNB together with EPC and HSS. This is the last step and now the LTE core network + eNB is started. First, required certificates are installed as follows:

```
$ cd ~/openair-cn/SCRIPTS
$ ./check_hss_s6a_certificate /usr/local/etc/oai/
freeDiameter/hss.openair4G.eur
$ ./check_mme_s6a_certificate /usr/local/etc/oai/
freeDiameter/example.openair4G.eur
```

NOTE: On the last line, 'example' need to match the chosen hostname/Fully Qualified Domain Name (FQDN) explained in Appendix A.

Next, the HSS is compiled and started. It is important to always run the HSS first:

```
$ cd ~/openair-cn
$ cd SCRIPTS
$ ./build_hss -c
$ ./run_hss -i ~/openair-cn/SRC/OAI_HSS/db/oai_db.sql
#Run only once to install database
$ ./run_hss #Run for all subsequent runs
```

Then, compile and run the MME followed by the SP-GW:

```
$ cd ~/openair-cn/SCRIPTS
$ ./build_mme -c
$ ./run_mme
```

```
$ cd ~/openair-cn
$ cd SCRIPTS
$ ./build_spgw -c
$ ./run_spgw
```

Finally, the eNB (USRP) is started, providing radio access to our OAI EPC for UEs near the base station:

```
$ cd ~/openairinterface5g
$ source oaienv
$ ./cmake_targets/build_oai -w USRP -x -c --eNB
$ cd cmake_targets/lte_build_oai/build
$ sudo -E ./lte-softmodem -O $OPENAIR_DIR/targets/PROJECTS
/GENERIC-LTE-EPC/CONF/enb.band7.tm1.usrpb210.conf -d
```

6. Activate a tcpdump session and save it to the desired location in form of a PCAP-file:

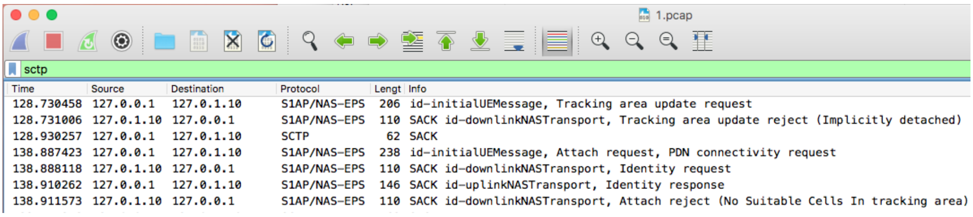
```
$ sudo tcpdump -i lo -w
/home/wlab/Desktop/docs/logs/capture.pcap
```

Now, all the necessary parts of the software are compiled and started. One green and one red light will be visible in the front of the USRP B200mini, indicating that all is working correctly. From this point, UEs inside the coverage area of the eNB will be able to see the mobile network and can connect to it.

4.4 My Own Results

4.4.1 Sending modified identity request

Firstly, to ensure that the setup and configurations of the IMSI Catcher were correct, it was tested with the original source code. Which implies that the identity request asks the UE to provide the IMSI for identification purposes. The whole IMSI Catcher was started as described in Section 4.3.3. Going into settings on the iPhone and choose to connect to the fake mobile network, the phone starts the attach procedure. The messages sent between the UE and mobile network can be seen in Figure 4.1. In this case, the UE (iPhone) has the IP address 127.0.0.1 and the network has 127.0.1.10. After the mobile sends the attach request message, the network replies with an identity request that by default will ask for the IMSI identity. Figure 4.2 shows details about the Identity request message sent by the mobile network. On the bottom with the blue outline, we acknowledge that the specified wanted identity indeed turns out to be IMSI.

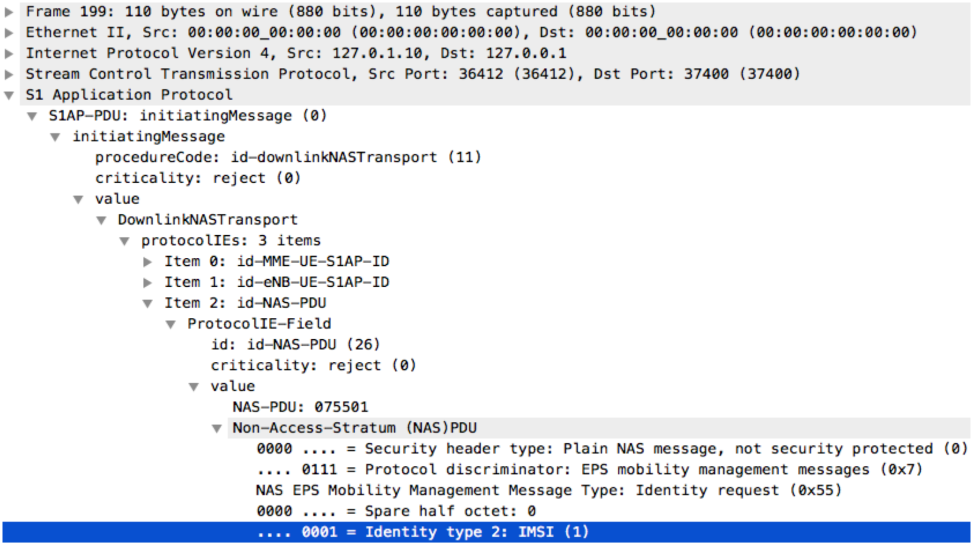


```
Time        Source      Destination   Protocol     Lengt Info
128.730458 127.0.0.1   127.0.1.10    S1AP/NAS-EPS  206 id-initialUEMessage, Tracking area update request
128.731006 127.0.1.10  127.0.0.1     S1AP/NAS-EPS  110 SACK id-downlinkNASTransport, Tracking area update reject (Implicitly detached)
128.930257 127.0.0.1   127.0.1.10    SCTP           62 SACK
138.887423 127.0.0.1   127.0.1.10    S1AP/NAS-EPS  238 id-initialUEMessage, Attach request, PDN connectivity request
138.888118 127.0.1.10  127.0.0.1     S1AP/NAS-EPS  110 SACK id-downlinkNASTransport, Identity request
138.910262 127.0.0.1   127.0.1.10    S1AP/NAS-EPS  146 SACK id-uplinkNASTransport, Identity response
138.911573 127.0.1.10  127.0.0.1     S1AP/NAS-EPS  110 SACK id-downlinkNASTransport, Attach reject (No Suitable Cells In tracking area)
```

Figure 4.1: UE connecting to the network (IMSI requested)

Just in the same way, the requested identity type was changed to IMEI, following the steps explained in Section 4.3.1. Figure 4.3 shows the captured identity request message that is sent to the UE. Considering that the requested identity type now is IMEI (seen on the last line in Figure 4.3), it can be concluded that the approach described in Section 4.3.1 works.

In comparison to the attempt using the original source code and requesting the IMSI, this attempt where the network asks the UE to provide its IMEI identity, results in a different exchange of messages between the UE and the eNB. Figure 4.4 shows the messages when the network asks for the IMEI. We can see that the identity response message is missing. The UE is not willing to answer the identity request message asking for the IMEI and ignores the multiple attempts from the network side. On the OAI website, they state that hosting both the EPC + eNB on the same computer can reduce real-time performance [19]. In step 4 of the attach procedure in Chapter 2, the timer T3470 is explained. Basically, this timer specifies how long the EPC must wait before re-transmitting the identity request message in case of no identity



```
▶ Frame 199: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 127.0.1.10, Dst: 127.0.0.1
▶ Stream Control Transmission Protocol, Src Port: 36412 (36412), Dst Port: 37400 (37400)
▼ S1 Application Protocol
  ▼ S1AP-PDU: initiatingMessage (0)
    ▼ initiatingMessage
        procedureCode: id-downlinkNASTransport (11)
        criticality: reject (0)
      ▼ value
        ▼ DownlinkNASTransport
          ▼ protocolIEs: 3 items
            ▶ Item 0: id-MME-UE-S1AP-ID
            ▶ Item 1: id-eNB-UE-S1AP-ID
            ▼ Item 2: id-NAS-PDU
              ▼ ProtocolIE-Field
                  id: id-NAS-PDU (26)
                  criticality: reject (0)
                ▼ value
                    NAS-PDU: 075501
                  ▼ Non-Access-Stratum (NAS)PDU
                      0000 .... = Security header type: Plain NAS message, not security protected (0)
                      .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
                      NAS EPS Mobility Management Message Type: Identity request (0x55)
                      0000 .... = Spare half octet: 0
                      .... 0001 = Identity type 2: IMSI (1)
```

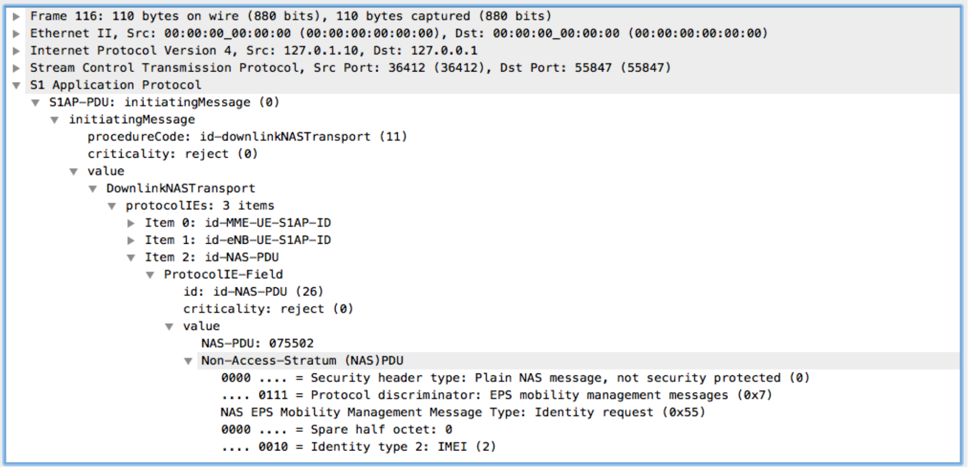
Figure 4.2: Identity request message (IMSI)

response from the UE. To ensure that delays in the communication between the UE and EPC are not the cause of no received identity response, different values for the T3470 timer was tested. The original value is 6 seconds and can be changed as explained in Appendix B. The experiment was then tested with T3470 values of 12 and 18 seconds. Still, the UE did not answer the network with an identity response.

4.4.2 Modified identity request with most significant bit = 1

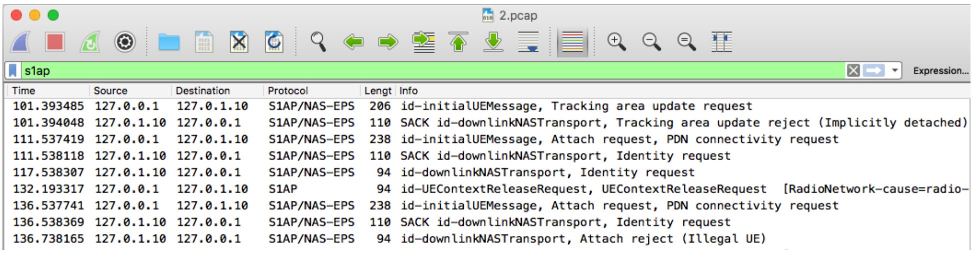
Since only changing the requested identity type to '2' (IMEI) did not work, the successful method from [33] will be tried. After the necessary code modifications as described in Section 4.3.2 was done, the different entities of the OAI software were rebuilt. Then, I connected the iPhone to the newly started mobile network. After some minutes, the IMSI Catcher was powered off as well as the TCP dump session. Figure 4.5 shows the captured identity request message sent from the mobile network to the UE.

Studying the information of Figure 4.5, the line on the bottom with blue outline shows that the most significant bit now actually is 1 (1010 = Identity type 2: Unknown (10)) and 4-bit encoding is working. Additionally, the corresponding decimal value equal to 10 is given in parentheses in the end of the line. Therefore, we are now sure that we have the same successful settings as used in the experiment from [33]. With that said, it seems like the trick with the most significant bit does not work



```
▶ Frame 116: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 127.0.1.10, Dst: 127.0.0.1
▶ Stream Control Transmission Protocol, Src Port: 36412 (36412), Dst Port: 55847 (55847)
▼ S1 Application Protocol
  ▼ S1AP-PDU: initiatingMessage (0)
    ▼ initiatingMessage
        procedureCode: id-downlinkNASTransport (11)
        criticality: reject (0)
      ▼ value
        ▼ DownlinkNASTransport
          ▼ protocolIEs: 3 items
            ▶ Item 0: id-MME-UE-S1AP-ID
            ▶ Item 1: id-eNB-UE-S1AP-ID
            ▼ Item 2: id-NAS-PDU
              ▼ ProtocolIE-Field
                  id: id-NAS-PDU (26)
                  criticality: reject (0)
                ▼ value
                    NAS-PDU: 075502
                  ▼ Non-Access-Stratum (NAS)PDU
                      0000 .... = Security header type: Plain NAS message, not security protected (0)
                      .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
                      NAS EPS Mobility Management Message Type: Identity request (0x55)
                      0000 .... = Spare half octet: 0
                      .... 0010 = Identity type 2: IMEI (2)
```

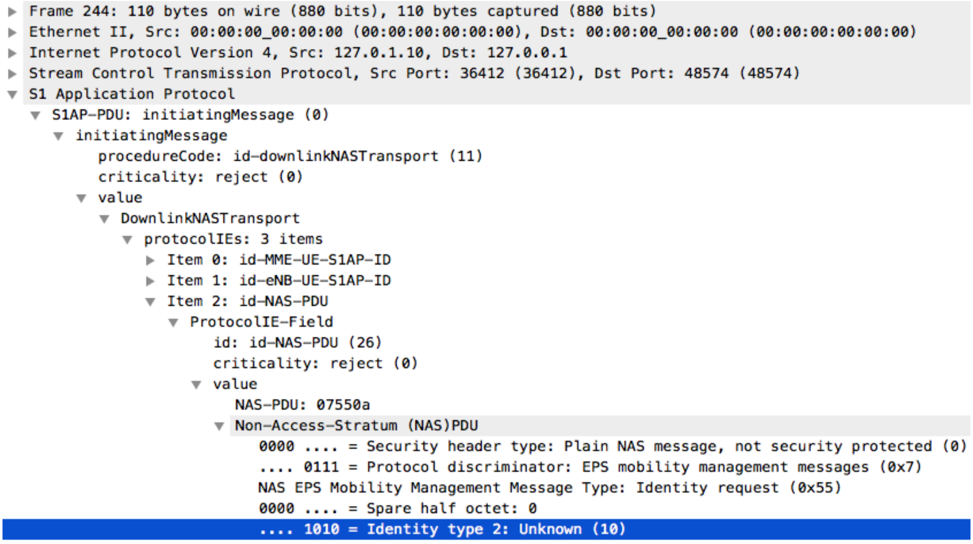
Figure 4.3: Identity request message (IMEI)



```
Time        Source      Destination  Protocol      Lengt  Info
101.393485  127.0.0.1   127.0.1.10   S1AP/NAS-EPS  206  id-initialUEMessage, Tracking area update request
101.394048  127.0.1.10  127.0.0.1    S1AP/NAS-EPS  110  SACK id-downlinkNASTransport, Tracking area update reject (Implicitly detached)
111.537419  127.0.0.1   127.0.1.10   S1AP/NAS-EPS  238  id-initialUEMessage, Attach request, PDN connectivity request
111.538118  127.0.1.10  127.0.0.1    S1AP/NAS-EPS  110  SACK id-downlinkNASTransport, Identity request
117.538307  127.0.1.10  127.0.0.1    S1AP/NAS-EPS   94  id-downlinkNASTransport, Identity request
132.193317  127.0.0.1   127.0.1.10   S1AP           94  id-UEContextReleaseRequest, UEContextReleaseRequest  [RadioNetwork-cause=radio-
136.537741  127.0.0.1   127.0.1.10   S1AP/NAS-EPS  238  id-initialUEMessage, Attach request, PDN connectivity request
136.538369  127.0.1.10  127.0.0.1    S1AP/NAS-EPS  110  SACK id-downlinkNASTransport, Identity request
136.738165  127.0.1.10  127.0.0.1    S1AP/NAS-EPS   94  id-downlinkNASTransport, Attach reject (Illegal UE)
```

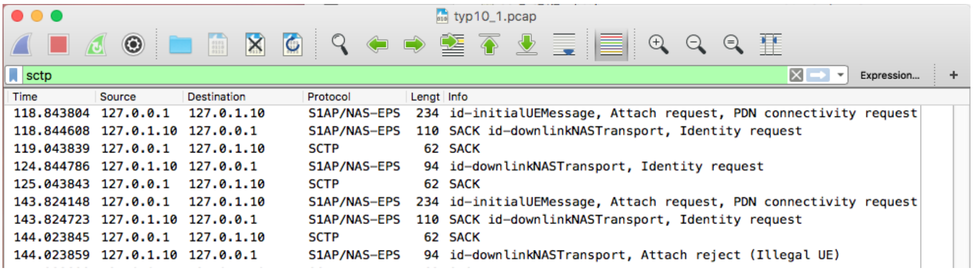
Figure 4.4: UE connecting to the network (IMEI requested)

in this case. As Figure 4.6 shows, the mobile network (IP address = 127.0.1.10) sends multiple identity request messages to the UE (IP address = 127.0.0.1) without receiving any identity response message from the UE. The same situation as when the requested identity type was set to 2 (IMEI). This experiment was repeated multiple times and no identity response messages were observed.



```
▶ Frame 244: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 127.0.1.10, Dst: 127.0.0.1
▶ Stream Control Transmission Protocol, Src Port: 36412 (36412), Dst Port: 48574 (48574)
▼ S1 Application Protocol
  ▼ S1AP-PDU: initiatingMessage (0)
    ▼ initiatingMessage
        procedureCode: id-downlinkNASTransport (11)
        criticality: reject (0)
      ▼ value
        ▼ DownlinkNASTransport
          ▼ protocolIEs: 3 items
            ▶ Item 0: id-MME-UE-S1AP-ID
            ▶ Item 1: id-eNB-UE-S1AP-ID
            ▼ Item 2: id-NAS-PDU
              ▼ ProtocolIE-Field
                  id: id-NAS-PDU (26)
                  criticality: reject (0)
                ▼ value
                    NAS-PDU: 07550a
                  ▼ Non-Access-Stratum (NAS)PDU
                      0000 .... = Security header type: Plain NAS message, not security protected (0)
                      .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
                      NAS EPS Mobility Management Message Type: Identity request (0x55)
                      0000 .... = Spare half octet: 0
                      .... 1010 = Identity type 2: Unknown (10)
```

Figure 4.5: Identity request message (most significant bit = 1)



```
Time        Source     Destination Protocol     Lengt Info
118.843804  127.0.0.1  127.0.1.10  S1AP/NAS-EPS  234  id-initialUEMessage, Attach request, PDN connectivity request
118.844608  127.0.1.10 127.0.0.1   S1AP/NAS-EPS  110  SACK id-downlinkNASTransport, Identity request
119.043839  127.0.0.1  127.0.1.10  SCTP           62  SACK
124.844786  127.0.1.10 127.0.0.1   S1AP/NAS-EPS   94  id-downlinkNASTransport, Identity request
125.043843  127.0.0.1  127.0.1.10  SCTP           62  SACK
143.824148  127.0.0.1  127.0.1.10  S1AP/NAS-EPS  234  id-initialUEMessage, Attach request, PDN connectivity request
143.824723  127.0.1.10 127.0.0.1   S1AP/NAS-EPS  110  SACK id-downlinkNASTransport, Identity request
144.023845  127.0.0.1  127.0.1.10  SCTP           62  SACK
144.023859  127.0.1.10 127.0.0.1   S1AP/NAS-EPS   94  id-downlinkNASTransport, Attach reject (Illegal UE)
```

Figure 4.6: UE connecting to the network (most significant bit = 1)

4.5 Discussion

Motivated by the extra identification information the IMEI can give about the UE as well as the successful experiment in the related work [33], a whole chapter was dedicated for the IMEI experiment. The fact that no academic papers in my knowledge have tried this using the OAI implementation of the EPS was also a motivation for this experiment. The experiment description shows that the needed change of code for manipulating the identity request procedure was relatively small. Interestingly, the results show that specifying the IMEI as wanted identity in the identity request procedure did not work. Captured traffic going between the connecting UE and the eNB shows that the identity request messages actually ask for the IMEI. Moreover, the length of the T3470 timer specifying how long the network shall wait for the identity response from the UE was tested up to triple as long as the standard value. Excluding the possibility of the identity response message being lost due to delays. Therefore, the possibility for wrong code manipulation inside the EPC in the experiment can most likely be excluded. The reason that the UE does not respond with an identity response containing its IMEI number, may be located on the UE side. This assumption is strengthened by the security specifications from 3GPP. The serving network may request the UE to send the IMEI of the terminal, but the UE shall only respond with the IMEI identity after the serving network has been authenticated with exception of emergency calls [2]. This can explain why the UE does not respond to the IMEI requests in this experiment since the serving network is not authenticated by the UE. Importantly, this result is good news for the privacy of LTE subscribers. On the other side, if the possibility of revealing the IMEI number depends on the modem contained in the UE, then the security may vary in different mobile phones. Therefore, it would be valuable to test a range of different UE types for the potential IMEI reveal vulnerability. Moreover, the specification document on how the UE should handle IMEI requests from the serving network states that the UE shall provide the IMEI during an emergency call [2]. Hypothetically, the emergency feature can be used to trick the modem in the UE to provide the IMEI without authenticating the serving network. However, as explained in step 1 in the LTE attach procedure from Chapter 2, the emergency mode is initiated by the UE. Consequently, it may be impossible to trigger the emergency mode feature from the attacker side, using an IMSI Catcher.

Chapter 5 MSISDN Catchers

5.1 Overview

Interestingly, there are not yet developed catchers for the MSISDN also known as a telephone number. The telephone number is used in a range of different contexts and can reveal much information about a person. This motivates for looking into the possibilities of an MSISDN Catcher. This chapter is going to answer the research questions (i) 'Is it possible to extends an IMSI Catcher to be an MSISDN Catcher?' and (ii) 'How strong is the MSISDN number protected?'. The context for these questions is a study of the security in the LTE standard regarding the MSISDN. Considering the LTE standard is relatively new, some shortcuts were made during the transition from UMTS to LTE. Based on the fact that LTE is purely IP-based, voice and data traffic would be transported over packet switched connections.

Many network operators that were eager to implement the LTE standard, used a solution called CSFB for voice transport in the beginning. Including Telenor in Norway [43]. Because the voice part of LTE named VoLTE was delayed, CSFB was used as a temporary solution. Basically, what CSFB does is downgrading the connection to UMTS or GSM when a call is made or received on an LTE-connected UE. In that way, the voice traffic is carried over the circuit switched architecture used in the older standards. More details about the CSFB can be found in the specification document at [12]. Consequently, the protection of the MSISDN in the UMTS standard most also be considered when analysing how good MSISDN is secured in LTE.

5.2 Related work

The analytical report with the title "Gone Opaque? An Analysis of Hypothetical IMSI Catcher Overuse in Canada" investigates the surveillance capabilities of IMSI Catchers [36]. The difficulty of retrieving the MSISDN number by using IMSI Catchers are discussed which makes it relevant for this chapter.

The paper 'Privacy Leaks in Mobile Phone Internet Access' studies if and how private mobile subscriber information is leaked through mobile phone-based web access [34]. Specifically, the paper focus on how private information is leaked from mobile phones through Hypertext Transfer Protocol (HTTP) headers. The author used his own website to log all the HTTP headers from the traffic generated by the mobile phone users connecting to the website. From the analysis of the logged traffic, they collected 1183 phone numbers. By studying the CC part of the collected MSISDN (explained in the LTE chapter), the nationality of the phone numbers can be identified. From the total collected phone numbers, they found that they came from 67 different countries. The fact that MSISDN actually was leaked through the HTTP headers makes this work relevant for this chapter.

The paper 'Detection of Side Channel Attacks Based on Data Tainting in Android Systems' analyses 100 Android applications for possible leakage of sensitive data through side channels [26]. Interestingly, they found that 35% of the applications leaked private information through side channels. Side channel attacks can potentially leak private info, including the MSISDN. These findings make the paper relevant for this chapter, as an alternative attack strategy compared to the principle of an IMSI Catcher.

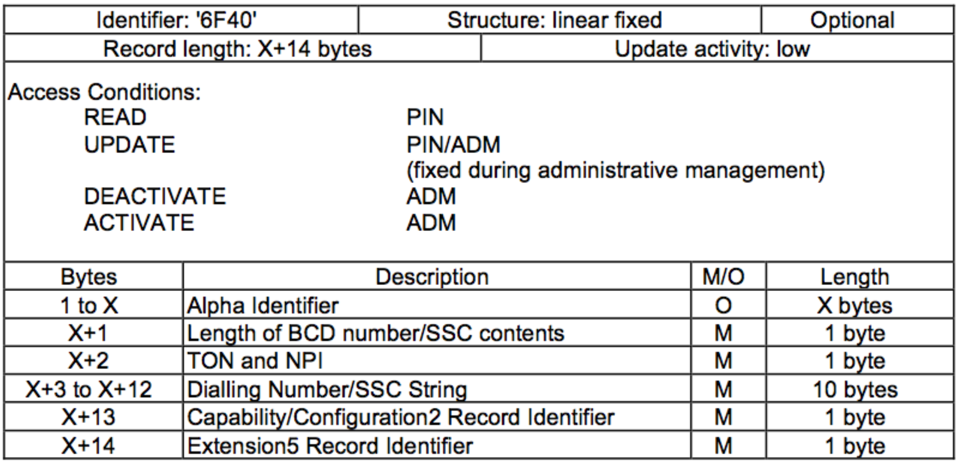
5.3 Security in LTE

This section looks at the security of the MSISDN in LTE. Specifically, the communication between the LTE and eNB will be of interest in cases where MSISDN is sent. Location of where the MSISDN is located and stored in the LTE architecture will also be presented.

5.3.1 The User Equipment

All UEs that want to enjoy services offered by the LTE network must identify and authenticate their subscription to the MNO. In the legacy networks such as GSM, this was done by sending subscription details stored on the Subscriber Identity Module (SIM) to the network. More specifically, the IMSI number securely stored on the SIM was sent to the network, and the responsible entity in the network searched for the provided IMSI number. The SIM used in GSM had multiple restrictions and the successors UMTS and LTE demanded a more complex smart card. The solution was the new removable Universal Integrated Circuit Card (UICC) smart card. Basically, the UICC is a removable smart card also. However, it has a microprocessor and a larger storage that can hold multiple applications. The two most common application located on the UICC is Universal Subscriber Identity Module (USIM) and IP Multimedia Services Identity Module (ISIM). To clarify, in GSM when talking about SIM, it meant the hardware and software of the smart card combined. In the new smart card used in UMTS and LTE, SIM refers only to the software. Specifically, the software part holding information used for identification and authentication was named USIM. The terminology of the hardware part is UICC. The ISIM application is used for access to IMS services [3]. Practically, the ISIM application also allows secure access to other independent services like payment.

TS 31.102 by 3GPP specifies that the MSISDN of the subscriber registered to the UICC can be stored in the USIM application [4] residing on the UICC. Figure 5.1 shows the Elementary File (EF) MSISDN contained in the USIM application. The EF can hold multiple MSISDNs related to the subscriber. In addition to the MSISDN, the file holds information such as identifiers of associated network bearer capabilities. Interestingly, the access restrictions on the EF for reading the MSISDN is set to PIN (see Figure 5.1). Which means that the user of the phone must type their PIN-code to read the MSISDN stored in the USIM application. Motivated by this, the next section will explain a way to read the MSISDN stored on the UICC on an iPhone running iOS 10.



Identifier: '6F40'	Structure: linear fixed	Optional
Record length: X+14 bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	PIN/ADM (fixed during administrative management)
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1 to X	Alpha Identifier	O	X bytes
X+1	Length of BCD number/SSC contents	M	1 byte
X+2	TON and NPI	M	1 byte
X+3 to X+12	Dialling Number/SSC String	M	10 bytes
X+13	Capability/Configuration2 Record Identifier	M	1 byte
X+14	Extension5 Record Identifier	M	1 byte

Figure 5.1: EF MSISDN description from [4]

Let us take a look at the EF MSISDN in Figure 5.1. The MSISDN itself is found in the Dialling Number or SSC String parameter with a length of 10 bytes. The Type of Number (TON) with length of 1 byte is the last byte before the MSISDN. Basically, as the name describes, the TON tells the type of the number. According to the GSM specification document 03.40 from [18], TON can have these values:

1. Unknown

2. International number
3. National number
4. Network specific number
5. Subscriber number
6. Alphanumeric number
7. Abbreviated number

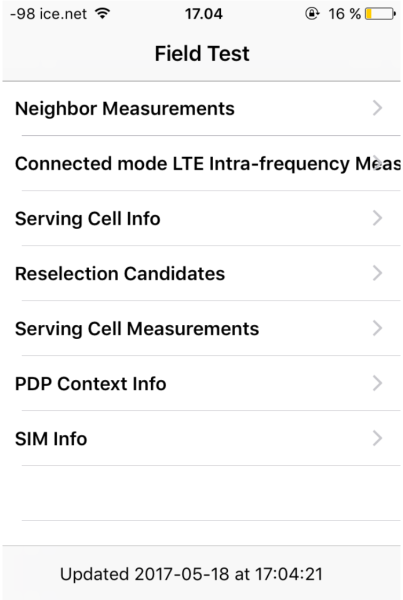
5.3.2 Field Test Feature on iOS

Today, the mobile equipment we carry around is capable of a range of tasks thanks to its advanced, powerful operating system residing on flexible hardware. Among all the nice functionality, some features are more hidden. One of these is the 'Field Test' feature. To activate this feature in iOS, do the following:

1. Open the phone app
2. Navigate to the keypad
3. Dial this number: *3001#12345#*
4. Press the call button

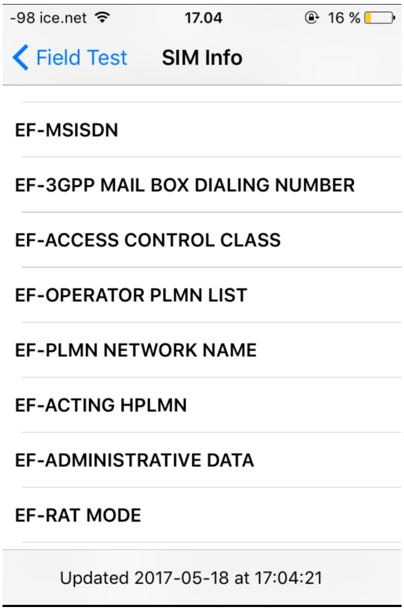
This should bring you to the menu shown in Figure 5.2.

In our case, we want to see if we can retrieve the MSISDN stored on the UICC. To do this, you shall navigate to the SIM info choice in the main menu in Figure 5.2. That will bring up the menu shown in Figure 5.3.



Updated 2017-05-18 at 17:04:21

Figure 5.2: Screenshot of the Field Test main menu on an iPhone 6s



Updated 2017-05-18 at 17:04:21

Figure 5.3: Screenshot of the SIM Info page showing the EF-MSISDN

Of privacy reasons the actual values of the different parameters are removed in Figure 5.3, but the MSISDN was actually readable in this menu on the line holding the 'EF-MSISDN' parameter. With that said, the MSISDN is stored in a special manner. Specifically, the format is like this: 'FFF....FFF'. Where the dots represent the fields listed in Figure 5.1. The MSISDN can be found in inverted Binary Coded Decimal (BCD) format. Subsequently, the conversion back to the default form can be illustrated by a short example:

A Norwegian telephone number is stored in the EF_MSISDN field located in the USIM application. Let the telephone number be +47 12345678. Table 5.1 shows how the telephone number is converted. Read from left to right, take two numbers standing by the side of each other, and switch them before moving on to the next two numbers to the right.

Standard form	USIM
47 12345678	74 21436587

Table 5.1: MSISDN format

From a security perspective, the MSISDN was relatively easy to retrieve from the UICC when you have access to the phone and the phone is powered on. The phone did not ask for the PIN-code during the Field Test procedure. This means that the USIM application authenticates the user for reading privileges after the PIN-code has been entered successfully when powering on the phone. After that, the only security mechanism protecting the information stored on the UICC including the MSISDN is the eventual personal password the user of the phone has activated. Considering that most regular users never will access this part of the UICC, it is few or none arguments for not requesting the PIN when using this special feature. That will add an additional layer of security, especially important in cases where the user does not have a personal password at all on the phone. With that said, this method of revealing the MSISDN requires the attacker to have physical access to the UE. Maybe, social engineering can be an option also, but it is no obvious way of using a modified IMSI Catcher to retrieve the MSISDN from this storage location.

5.4 Security of the MSISDN in VoLTE

Generally, the way we use the telephone number most of the time are when we make a voice call or send an SMS. This is done by either dialling the telephone number directly or by finding the telephone number in the address book, before pressing the call button. Then the call setup procedures are initiated and the called party is notified by an incoming call showing the MSISDN of the call originator. In LTE, the VoLTE technology makes this possible. As explained in the IMS Section in Chapter 2, UEs must initiate the IMS registration process after successful LTE attachment, to be available for IMS services. Interestingly, the IMS registration procedure involves the establishment of IPSec encryption of all signalling traffic between the UE and the EPC network. Consequently, interception of VoLTE traffic holding potential public user identities (explained in Chapter 2) using a modified IMSI Catcher, becomes very difficult or even impossible. Moreover, during authentication setup in the IMS registration procedure, information stored in the ISIM application residing on the UICC is needed. In addition to hold authentication related information, it also contains public and private identities tied to the subscriber (MSISDN). Seen from a security perspective, it is no obvious way of query the MSISDN residing in the ISIM application on the UE using a modified IMSI Catcher either. This is because, in comparison to catching IMSI or IMEI where a modified identity request procedure can be exploited, there is no such equivalent procedure for asking the UE to provide the MSISDN to the eNB. Briefly, in a situation where a procedure in the EPC needs to address a connecting subscriber to an MSISDN, the IMSI identity of the subscriber is used as a key to query the HSS database for the MSISDN.

5.5 Leaks of private information in HTTP headers

An interesting approach for catching private information from MSs is through the HTTP headers. This is made possible as a consequence of the HTTP Header Enrichment functionality. According to the related work [34], private information leakage through HTTP headers is related to HTTP proxies operated by MNOs. When the subscriber is browsing the web using their MS over a mobile network connection, these proxies inject additional headers into the active HTTP connections. Including information of private nature. The way an attacker can exploit this situation is by the help of a website that has visitors browsing the website from their MS. In that way, their MSISDN can be injected into additional headers, which are sent together with the standard HTTP traffic to the website controlled by the attacker. If the attacker then logs the HTTP connections to the website, they can be analysed and potential private information of the visitors can be extracted from the headers added by the proxies. Consequently, a reverse lookup can be done on the captured MSISDN numbers by searching for them in online phone books. Norway has multiple public websites offering possibilities for such reverse look-ups, mapping telephone numbers to the name of the owner[1,2]. Additionally, by dismantling the captured MSISDN numbers into the building blocks explained in Chapter 2, the CC part of the MSISDN can be used to track visitors of the website to the country where their subscriptions originate from.

The results of the experiment conducted by Mulliner acknowledges a serious privacy leakage in HTTP headers. By using his own website, all HTTP headers that originated from visitors was logged. Later, the logs were analysed with the focus on revealing telephone numbers. The work resulted in collection of 1183 phone numbers from 67 different countries shown in figure 5.4

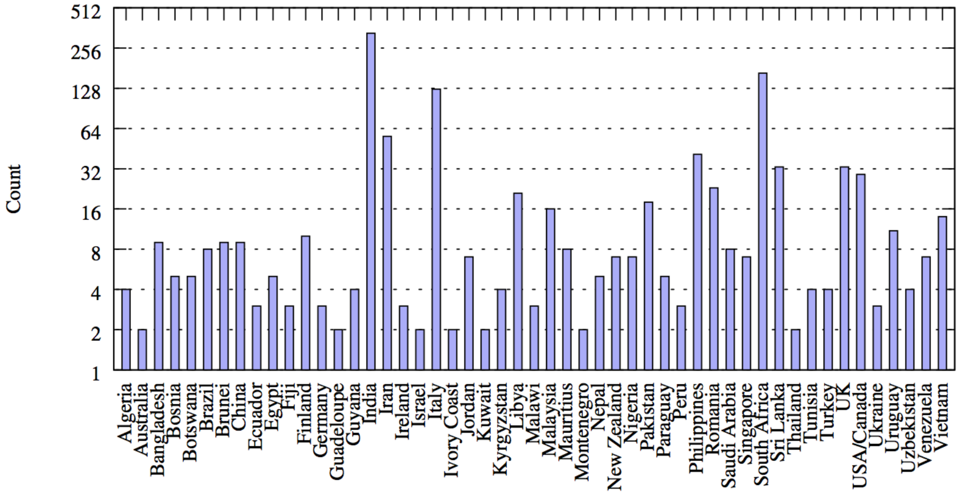


Figure 5.4: Showing collected MSISDN by nationality. From [34]

[1] https://www.1881.no

[2] https://www.gulesider.no

5.6 Discussion

The way we use telephone numbers today has definitely evolved from the beginning. In addition to identification during call setup, the telephone number is often used in eg. mobile authentication where a secret code is sent to the phone number. Also, phone numbers are often used as identification for additional mobile services and often presented on social media. As a consequence of the huge digitisation during the past decades on many platforms, services previously offered in written format is now available online offering efficient searching mechanisms. E.g. the traditionally physical telephone books holding the telephone numbers to all subscribers in a geographic area was eventually digitised and published online. Gulesider and 1881 are examples of online telephone books in Norway[3,4]. Importantly, this trends of pushing increasing amounts of information online implicate a change of vulnerabilities and threats. With online telephone books presenting the affiliation between MSISDN and the subscriber, a reverse lookup can in many cases be conducted on caught MSISDNs with little effort. In that way, the attacker can easily find the owner of the telephone number. Further, the name of the subscriber can again be used in searches towards social media, institutions, news and more. Dramatically, the ability of catching the phone numbers of UEs near the base station, will expand the amount of privacy invasion channels available for unlawful attackers. Earlier, these information channels were restricted to the authorities and the MNOs because reverse lookup services on IMSI numbers are not available to the public.

The security of LTE is complex. Focusing on the MSISDN, relevant procedures and location of the MSISDN is important to cover. Primarily, the MSISDN is stored centralised inside the HSS located in the EPC together with other important subscriber information. Basically, this means that the MSISDN stored in the HSS is unreachable from the traditional IMSI Catcher where information flowing to and from the eNB may be revealed. Therefore, the IMSI Catcher must look for the MSISDN another place. Actually, the MSISDN can be stored in the UICC inside the UE. However, storage of the MSISDN is optional. Also, the owner of the UE can by specification change the stored MSISDN in the USIM application, making the reliability questionable. Importantly, the read operation implemented in the USIM application is protected by the PIN-code. However, the experiment using the field test feature demonstrated that if a user turns on the phone and types the PIN, the user is authenticated to read the MSISDN from the UICC at all times until the phone is turned off again. Notably, this applies only to the tested mobile phone (iPhone 6s running iOS 10). Compared to catching the IMEI or IMSI by exploiting the identity request procedure, there is no such procedure for retrieving MSISDN optionally stored on the USIM. Consequently, retrieving the MSISDN from the UE

[3] https://www.gulesider.no

[4] https://www.1881.no

using a traditional IMSI Catcher may be impossible. Hence, the only way to reveal the MSISDN by going after the stored value on the UICC is by physical access to a powered on mobile telephone with no personal password activated. Thus, this storage location of the MSISDN may seem less important to secure since the only exploit possibility is by physical access to the phone. With that said, permissions for subscriber related information stored in UICC applications should be protected. As found in the paper 'Detection of Side Channel Attacks Based on Data Tainting in Android Systems', mobile applications can leak private information through side channels, including the MSISDN [26]. On one hand, the traditional IMSI Catcher may not be able to retrieve the MSISDN directly from the UICC. On the other hand, a mobile application specified to leak the MSISDN through side channels back to the developer can be a possibility.

Moving over to the network procedures involving the MSISDN, voice communication is essential. The solution for voice communication in LTE is mainly provided by VoLTE or CSFB. Thinking of the behaviour of an IMSI Catcher, the ability of monitoring or intercepting the traffic going between the UE and eNB seems like a good strategy. Specifically, intercepting a voice call during setup and then try to extract the MSISDN. However, all VoLTE subscribers are forced to initiate the IMS registration procedure after the ordinary LTE attach. Involving establishment of IPSec encryption of all subsequent communication. Which makes it difficult to decrypt potential captured voice traffic. If a call is made on a mobile network which has implemented CSFB technology for voice communication, then the connection falls back to solutions implement in legacy networks. The needed signalling inside the EPS to support CSFB are actually protected by the LTE security mechanisms [23]. However, the security of the legacy part that actually offers the CS voice service, may not be secured in the same way as inside the EPS. Interestingly, the act of browsing the Internet over HTTP can reveal the MSISDN of the MS. The way HTTP proxies operated by MNOs, inject additional HTTP headers holding private information is known as the HTTP header enrichment feature. Basically, MNO uses this feature for operational purposes, but also to assist advertising programs to identify subscribers responsible for generating the traffic [44]. This is a serious privacy concern and the HTTP traffic can by this technology be linked to the telephone number of the visitor. Subsequently, side channels of information previously discussed, can be used to harvest even more detailed information about the visitor. By using the more secure communication protocol Hypertext Transfer Protocol Secure (HTTPS), the problem is solved. In order to perform header injection in HTTPS traffic, the Internet Service Provider (ISP) have to execute Transport Layer Security (TLS) interception of the traffic [44]. After all, the subscriber should be aware of the private information that may be leaked from HTTP traffic, originating from their phones.

Considering the first research question defined in chapter 1; "Can an IMSI Catcher be extended to catch MSISDN?", the findings point in another direction.

Based on findings such as no implemented request procedure for MSISDN, IPSec encryption initiated during IMS registration, and no possibility for retrieving the MSISDN from the USIM nor the HSS location remotely, it can be argued that the idea of using an IMSI Catcher to catch the MSISDN may be difficult. As identified by the vulnerabilities found in HTTP header enrichment technology, and privacy leakage through side channels in Android applications, there are alternatives not involving the idea of an IMSI Catcher. Combining such types of technological attacks, together with social engineering targeting services that include the MSISDN seems like the best strategy. Hence, further work may look more into the possibility of a mobile application dedicated to leaking private information such as the MSISDN to the attacker. The application should be tested on multiple mobile operating systems, to cover OS dependent vulnerabilities.

Chapter 6 Conclusion

The possibilities of building an "MSISDN-Catcher" has been investigated in this master thesis. A technical chapter about the LTE architecture, identification entities, relevant procedures and the IP Multimedia Subsystem has been presented.

Software and tools used for the IMEI experiment were shown in Chapter 3.

In Chapter 4 an IMSI Catcher based on the OpenAirInterface open source software and USRP B200mini radio device was built. The purpose of the experiment was to reveal the IMEI number of mobile phones camping near the base station. Potentially, being a valuable contribution to the MSISDN Catcher. A thorough explanation about needed changes of the OpenAirInterface source code was presented as well as the required steps to setup and configure the IMSI Catcher. From the experiment, it was found that the UE did not respond to identity requests asking for the IMEI identity. Which harmonise with the 3GPP specified behaviour of the UE in such situations. Also, the successful method for revealing the IMEI explained in [33] was tested. It was found that the trick did not work in this experiment. After examining the captured messages sent between the eNB and the UE, it can be concluded that the network actually asked for the IMEI. The evidence suggests that the modem located in the UE was responsible for ignoring the received IMEI request. For that reason, bad implementations found in modems of some UEs can explain why some of them reveal their IMEI to the network.

Chapter 5 investigates the security of where the telephone number is stored in the network, as well as the procedures that include the MSISDN. It was found that the MSISDN primarily was stored in the HSS. Optionally, the MSISDN could be stored in the USIM application located on the UICC. Besides, it was found that there is no implemented request procedure meant for asking the UE about providing the MSISDN to the network. Another essential point was that IMS initiate encryption of traffic during the mandatory IMS registration process. As a result, revealing MSISDNs by interception of voice calls (VoLTE) becomes very difficult. These findings contribute to the conclusion that it is no obvious way of using the principle of an IMSI Catcher to build an MSISDN Catcher.

Despite this, related work shows that MSISDN can be leaked in HTTP headers and through side channels of Android applications. Furthermore, it was shown how to read the MSISDN from the USIM application using the field test feature in iOS. Considering these findings, it can be concluded that there are possibilities for building an MSISDN Catcher. In summary, this master thesis found that the MSISDN and IMEI are well protected against open source LTE IMSI Catchers. However, the secrecy of the telephone number may not resist social attacks nor mobile applications exploiting information leakage through side channels.

6.1 Further Work

The outcome of the conclusion are interesting questions regarding the potential vulnerabilities in side channels of mobile applications as well as creative social attacks targeting the telephone number. Hence, further work may look more into the possibility of a mobile application dedicated to leaking private information such as the MSISDN to the attacker. The mobile application should be tested on multiple mobile operating systems, to cover eventual OS dependent vulnerabilities.

References

[1] 3GPP. Industry backs VoLTE initiative. http://www.3gpp.org/news-events/partners-news/1285-Industry-backs-VoLTE-initiative, 2010. [Online; accessed 27-May-2017].

[2] 3GPP. 3G security; Security architecture. TS 33.102, 3rd Generation Partnership Project (3GPP), March 2017.

[3] 3GPP. Characteristics of the IP Multimedia Services Identity Module (ISIM) application. TS 31.103, 3rd Generation Partnership Project (3GPP), 4 2017.

[4] 3GPP. Characteristics of the Universal Subscriber Identity Module (USIM) application. TS 31.102, 3rd Generation Partnership Project (3GPP), 3 2017.

[5] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2. TS 36.300, 3rd Generation Partnership Project (3GPP), April 2017.

[6] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception. TS 36.101, 3rd Generation Partnership Project (3GPP), April 2017.

[7] 3GPP. General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access. TS 23.401, 3rd Generation Partnership Project (3GPP), March 2017.

[8] 3GPP. IP Multimedia Subsystem (IMS); Stage 2. TS 23.228, 3rd Generation Partnership Project (3GPP), 3 2017.

[9] 3GPP. Network architecture. TS 23.002, 3rd Generation Partnership Project (3GPP), March 2017.

[10] 3GPP. Numbering, addressing and identification. TS 23.003, 3rd Generation Partnership Project (3GPP), March 2017.

[11] 3GPP. Policy and charging control architecture. TS 23.203, 3rd Generation Partnership Project (3GPP), March 2017.

[12] 3GPP. Technical Specification Group Services and System Aspects; Circuit Switched (CS) fallback in Evolved Packet System (EPS); Stage 2 (Release 14). TS 23.272, 3rd Generation Partnership Project (3GPP), March 2017.

[13] GSM Association et al. Volte service description and implementation guidelines version 2.0, 2014.

[14] Christopher Cox. *An introduction to LTE: LTE, LTE-advanced, SAE and 4G mobile communications.* John Wiley & Sons, 2012.

[15] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, and Edgar Weippl. Imsi-catch me if you can: Imsi-catcher-catchers. In *Proceedings of the 30th annual computer security applications Conference*, pages 246–255. ACM, 2014.

[16] J. Dong. *Network Dictionary.* ITPro collection. Javvin Technologies, Incorporated, 2007. page 252.

[17] Ericsson. World's first 4G/LTE network goes live today in Stockholm. https://www.ericsson.com/en/press-releases/2009/12/1360881-worlds-first-4glte-network-goes-live-today-in-stockholm, 2009. [Online; accessed 8-june-2017].

[18] GSM ETSI. 03.40. *Digital cellular telecommunications system (Phase 2+)*, 1.

[19] Eurocom. How to Connect OAI eNB (USRP B210) with COTS UE. https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/HowToConnectCOTSUEwithOAIeNBNew, 2017. [Online; accessed 21-May-2017].

[20] Eurocom. Openair Kernel Main Setup. https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirKernelMainSetup, 2017. [Online; accessed 30-May-2017].

[21] Eurocom. Openair System Requirements. https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirSystemRequirements, 2017. [Online; accessed 21-May-2017].

[22] Eurocom. Welcome to the OpenAirInterface project. https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/home, 2017. [Online; accessed 21-May-2017].

[23] Dan Forsberg, Günther Horn, Wolf-Dietrich Moeller, and Valtteri Niemi. Security for voice over lte. *LTE Security, Second Edition*, pages 215–232, 2010.

[24] 3GPP MCC Frédéric Firmin. NAS. http://www.3gpp.org/technologies/keywords-acronyms/96-nas, 2017. [Online; accessed 20-April-2017].

[25] 3GPP MCC Frédéric Firmin. The Evolved Packet Core. http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core, 2017. [Online; accessed 29-May-2017].

[26] Mariem Graa, Nora Cuppens-Boulahia, Frédéric Cuppens, Jean-Louis Lanet, and Routa Moussaileb. Detection of side channel attacks based on data tainting in android systems. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 205–218. Springer, 2017.

[27] GSMA. IR.88 LTE and EPC roaming guidelines V14.0. http://www.gsma.com/newsroom/all-documents/httpwww-gsma-comnewsroomwp-contentuploadsir-88-v14-0-pdf/, 2016. [Online; accessed 27-May-2017].

[28] GSMA. IR.92 IMS profile for voice and SMS V10.0. http://www.gsma.com/newsroom/all-documents/ir-92-ims-profile-for-voice-and-sms/, 2016. [Online; accessed 27-May-2017].

[29] GSMA. GSMA Intelligence Global Data. https://www.gsmaintelligence.com, 2017. [Online; accessed 7-june-2017].

[30] ITU-T. The international identification plan for public networks and subscriptions. Recommendation E.212, ITU Telecommunication Standardization Sector, September 2016.

[31] Nasjonal kommunikasjonsmyndighet. Nummerplan: E.212. https://www.nkom.no/npt/numsys/E.212.pdf, 2017. [Online; accessed 25-April-2017].

[32] Benoir Michau. Minimal LTE / EPC core network. https://github.com/mitshell/corenet, 2016. [Online; accessed 20-April-2017].

[33] Benoit Michau and Christophe Devine. How to not break LTE crypto. https://www.sstic.org/media/SSTIC2016/SSTIC-actes/how_to_not_break_lte_crypto/SSTIC2016-Article-how_to_not_break_lte_crypto-michau_devine.pdf, 2016. [Online; accessed 20-April-2017].

[34] Collin Mulliner. Privacy leaks in mobile phone internet access. In *Intelligence in Next Generation Networks (ICIN), 2010 14th International Conference on*, pages 1–6. IEEE, 2010.

[35] RCR Wireless News. LTE Attach Procedure Call Flow. http://www.rcrwireless.com/20140509/wireless/lte-attach-procedure-call-flow, 2014. [Online; accessed 11-May-2017].

[36] Christopher A Parsons and Tamir Israel. Gone opaque? an analysis of hypothetical imsi catcher overuse in canada. 2016.

[37] Andre Perez. *Voice over LTE: EPS and IMS networks*. John Wiley & Sons, 2013.

[38] Miikka Poikselkä, Harri Holma, Jukka Hongisto, Juha Kallio, and Antti Toskala. *Voice over LTE (VoLTE)*. John Wiley & Sons, 2012.

[39] Ettus Reseach. USRP B200mini. https://www.ettus.com/product/details/USRP-B200mini, 2017. [Online; accessed 13-May-2017].

[40] Torjus Bryne Retterstøl. Base station security experiments using usrp. Master's thesis, NTNU, 2015.

[41] SMScarrier.EU. Mobile Country Codes (MCC) and Mobile Network Codes (MNC). http://www.mcc-mnc.com, 2017. [Online; accessed 2-May-2017].

[42] SPIRENT. IMS Architecture - The LTE User Equipment Perspective. https://www.spirent.com/~/media/White%20Papers/Mobile/IMS_Architecture_White_Paper.pdf, 2014. [Online; accessed 27-May-2017].

[43] tek.no. Telenor åpner 4G for mobiltelefoner. https://www.tek.no/artikler/telenor-apner-4g-for-mobiltelefoner/115029, 2012. [Online; accessed 21-April-2017].

[44] Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, and Vern Paxson. Header enrichment or isp enrichment?: Emerging privacy threats in mobile networks. In *Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, pages 25–30. ACM, 2015.

[45] Wireshark. Wireshark. https://www.wireshark.org, 2017. [Online; accessed 10-june-2017].

[46] RF Wireless World. LTE tutorial. http://www.rfwireless-world.com/Tutorials/LTE-tutorial.html, 2017. [Online; accessed 12-May-2017].

Appendix A

OAI Installation

A.1 Kernel Configurations

Before installing the OAI software, the system requirements presented in chapter 3 must be fulfilled. Then the right OS must be installed. In this thesis, the standard 64-bit Ubuntu 14.04 version is used. After the installation is done, the correct kernel configuration must be applied. The kernel version used is kernel 3.19 low-latency. The kernel is installed as follows:

```
$ sudo apt-get install linux-image-3.19.0-61-lowlatency
linux-headers-3.19.0-61-lowlatency
```

To check that the installation was correct, restart the computer and the command 'uname -a' in the terminal should give the following output:

```
Linux [NAME] 3.19-lowlatency #201408132253 SMP PREEMPT Thu Aug
14 03:01:44 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
```

Moreover, configurations related to power management must be checked. Power management features located in BIOS, CPU frequency scaling, and hyperthreading must be turned off. The instructions for how to disable frequency scaling in Linux can be found at the askubuntu forum[1]. p-state and c-state must be disabled in Linux by changing the boot options located in the file path "/etc/default/grub" to this:

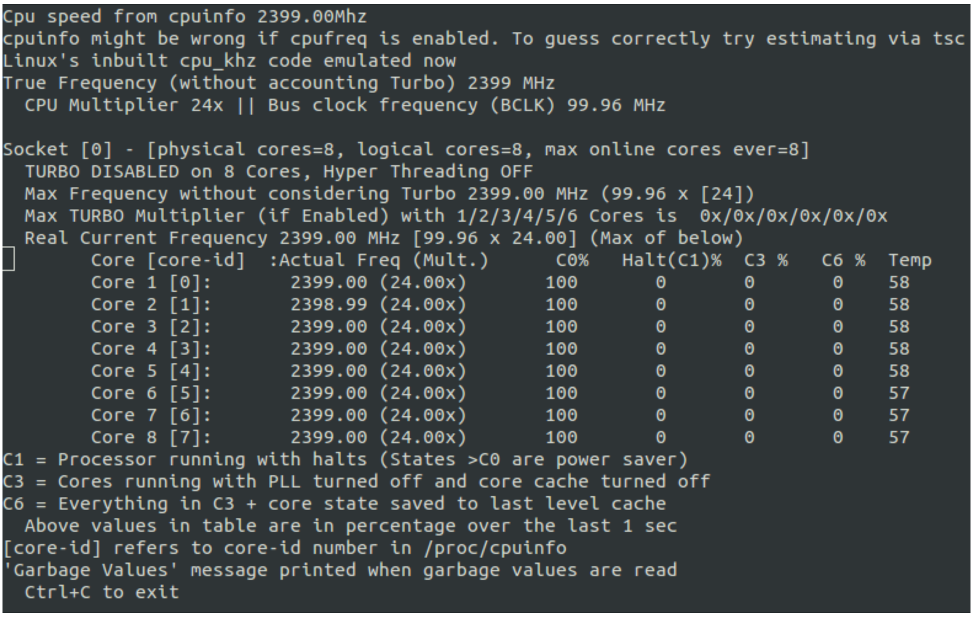
[1] https://askubuntu.com/questions/523640/how-i-can-disable-cpu-frequency-scaling-and-set-the-system-to-performance

```
$ GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_pstate=disable"
$ "processor.max_cstate=1 intel_idle.max_cstate=0 idle=poll"
```

Then run the command "update-grub". Further, the intel__powerclamp module must be blacklisted by appending "blacklist intel__powerclamp" to the end of the blacklist.conf file located in /etc/modprobe.d/. Finally, hyperthreading, CPU frequency control, P-states and C-states are disabled from BIOS. To check that the disablement of the power management features was successful, a CPU utility named i7z can be useful. Do the following to install i7z:

```
$ sudo apt-get install i7z
$ sudo i7z
```

The output from i7z is shown in figure A.1. Check that hyperthreading is off, that only C-state C0 is active and that the CPU do not change its frequency with more than 1-2 Hz



```
Cpu speed from cpuinfo 2399.00Mhz
cpuinfo might be wrong if cpufreq is enabled. To guess correctly try estimating via tsc
Linux's inbuilt cpu_khz code emulated now
True Frequency (without accounting Turbo) 2399 MHz
  CPU Multiplier 24x || Bus clock frequency (BCLK) 99.96 MHz

Socket [0] - [physical cores=8, logical cores=8, max online cores ever=8]
  TURBO DISABLED on 8 Cores, Hyper Threading OFF
  Max Frequency without considering Turbo 2399.00 MHz (99.96 x [24])
  Max TURBO Multiplier (if Enabled) with 1/2/3/4/5/6 Cores is  0x/0x/0x/0x/0x/0x
  Real Current Frequency 2399.00 MHz [99.96 x 24.00] (Max of below)
        Core [core-id]  :Actual Freq (Mult.)      C0%   Halt(C1)%  C3 %   C6 %  Temp
        Core 1 [0]:       2399.00 (24.00x)       100       0       0       0    58
        Core 2 [1]:       2398.99 (24.00x)       100       0       0       0    58
        Core 3 [2]:       2399.00 (24.00x)       100       0       0       0    58
        Core 4 [3]:       2399.00 (24.00x)       100       0       0       0    58
        Core 5 [4]:       2399.00 (24.00x)       100       0       0       0    58
        Core 6 [5]:       2399.00 (24.00x)       100       0       0       0    57
        Core 7 [6]:       2399.00 (24.00x)       100       0       0       0    57
        Core 8 [7]:       2399.00 (24.00x)       100       0       0       0    57
C1 = Processor running with halts (States >C0 are power saver)
C3 = Cores running with PLL turned off and core cache turned off
C6 = Everything in C3 + core state saved to last level cache
  Above values in table are in percentage over the last 1 sec
[core-id] refers to core-id number in /proc/cpuinfo
'Garbage Values' message printed when garbage values are read
  Ctrl+C to exit
```

Figure A.1: i7z output from [20]

The last step in the kernel configuration process is to disable CPU frequency scaling in Linux. Install the tool cpufrequtils for this task:

```
    $ sudo apt-get install cpufrequtils
```

Followed by adding the line GOVERNOR="Performance" in cpufrequtils located in /etc/default/ and save. To avoid that these modifications are overwritten during next reboot of the computer, disable on demand daemon by typing the following in the terminal:

```
    $ sudo update-rc.d ondemand disable
```

A.2 Download OAI software

After installing Ubuntu 14.04 64-bit with the right kernel configurations, the next step is to retrieve the OAI source code from their gitLab server[2].

The repository named Ooenairinterface5g contains the source code for the implementation of eNB RAN and UE RAN. The openair-cn repository holds the source code of the EPC part. To download the repositories from the gitLab server, go to the terminal and type the following:

```
    $ git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git
    $ git git clone https://gitlab.eurecom.fr/oai/openair-cn.git
```

A.3 Installation of OAI EPC and eNB

The first step is to check and eventually change the FQDN specified in the file 'hostname' located in the folder /etc/ on Linux. To edit the FQDN, open the 'hostname'-file using the Linux command-line text editor Nano. The first and only word in this field, represent the hostname of the computer:

```
    $ sudo nano /etc/hostname
```

[2] https://gitlab.eurecom.fr/oai

Change the hostname to the one you want and then save the file. The computer may need to restart, before the updated hostname/FQDN take effect [19]. Further, the correct FQDN must be updated in the file /etc/hosts. Let us say that the correct FQDN is 'example'. Then the correct content of the /etc/host file should be:

```
127.0.0.1    localhost
127.0.1.1    example.openair4G.eur   example
127.0.1.1    hss.openair4G.eur    hss
```

Appendix B

OAI eNB Machine Configuration

This appendix, presents the correct configuration to use in an experimental setup running eNB and EPC on the same host computer. After the installation steps explained in appendix A, configuration files included in the source code must be updated. First, check that the right parameters are given in the eNB configuration file located in the "openairinterface5g" folder (~/openairinterface5g/targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tm1.usrpb210.conf):

```
tracking_area_code  =  "1";
mobile_country_code =  "208";
mobile_network_code =  "92";



////////// MME parameters:
    mme_ip_address      = ( { ipv4       = "127.0.1.10";
                              ipv6       = "192:168:30::17";
                              active     = "yes";
                              preference = "ipv4";
                            }
                          );
    NETWORK_INTERFACES :
    {
        ENB_INTERFACE_NAME_FOR_S1_MME         = "lo";
        ENB_IPV4_ADDRESS_FOR_S1_MME           = "127.0.1.2/8";

        ENB_INTERFACE_NAME_FOR_S1U            = "lo";
        ENB_IPV4_ADDRESS_FOR_S1U              = "127.0.6.2/8";
        ENB_PORT_FOR_S1U                      = 2152; # Spec 2152
    };
```

The three top lines are important for the experimental configurations. Here, the TAC, MCC and MNC is configured. mme_ip_address shall be configured to the IP address of the network interface of the OAI EPC. NETWORK_INTERFACES holds network interface details of the OAI eNB.

The next step is to copy the configuration files provided in the OAI software over to local storage on the host computer (/usr/local/etc/oai):

```
$ sudo mkdir -p /usr/local/etc/oai/freeDiameter
$ sudo cp ~/openair-cn/ETC/mme.conf /usr/local/etc/oai
$ sudo cp ~/openair-cn/ETC/hss.conf /usr/local/etc/oai
$ sudo cp ~/openair-cn/ETC/spgw.conf /usr/local/etc/oai
$ sudo cp ~/openair-cn/ETC/acl.conf /usr/local/etc/oai/freeDiameter
$ sudo cp ~/openair-cn/ETC/mme_fd.conf /usr/local/etc/oai
/freeDiameter
$ sudo cp ~/openair-cn/ETC/hss_fd.conf /usr/local/etc/oai
/freeDiameter
```

The configuration for the MME (/usr/local/etc/oai/mme.conf) should look similar to this:

```
REALM = "openair4G.eur";

    S6A :
    {
      S6A_CONF     = "/usr/local/etc/oai/freeDiameter/mme\_fd.conf";
      HSS_HOSTNAME = "hss";
    };

GUMMEI_LIST = (
        {MCC="208" ; MNC="92"; MME_GID="4" ; MME_CODE="1"; }
     );

TAI_LIST = (
{MCC="208" ; MNC="92";  TAC = "1"; }
);

   NETWORK_INTERFACES :
    {
        MME_INTERFACE_NAME_FOR_S1_MME         = "lo";
        MME_IPV4_ADDRESS_FOR_S1_MME           = "127.0.1.10/8";

        # MME binded interface for S11 communication (GTPV2-C)
        MME_INTERFACE_NAME_FOR_S11_MME        = "lo";
        MME_IPV4_ADDRESS_FOR_S11_MME          = "127.0.8.11/8";
        MME_PORT_FOR_S11_MME                  = 2123;
    };

S-GW :
{
    # S-GW binded interface for S11 communication (GTPV2-C),
    if none selected the ITTI message interface is used
    SGW_IPV4_ADDRESS_FOR_S11                = "127.0.8.1/8";
};
```

GUMEI_LIST and TAI_LIST holds the MCC and MNC. When specifying the MCC and MNC, both the MME configuration file (mme.conf) and the eNB configuration file (enb.band7.tm1.usrpb210.conf) must be updated with the correct values.

The correct values for SPGW configuration (/usr/local/etc/oai/spgw.conf):

```
S-GW :
{

    NETWORK_INTERFACES :
    {
        SGW_INTERFACE_NAME_FOR_S11      = "lo";
        SGW_IPV4_ADDRESS_FOR_S11        = "127.0.8.1/8";


        SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP    = "lo";
        SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP      = "127.0.6.1/8";
        SGW_IPV4_PORT_FOR_S1U_S12_S4_UP         = 2152;

        # S-GW binded interface for S5 or S8 communication
        SGW_INTERFACE_NAME_FOR_S5_S8_UP    = "none";
        SGW_IPV4_ADDRESS_FOR_S5_S8_UP      = "0.0.0.0/24";
    };

...
}


P-GW =
{
    NETWORK_INTERFACES :
    {
        # P-GW binded interface for S5 or S8 communication
        PGW_INTERFACE_NAME_FOR_S5_S8          = "none";
        PGW_IPV4_ADDRESS_FOR_S5_S8            = "0.0.0.0/24";

        # P-GW binded interface for SGI
        PGW_INTERFACE_NAME_FOR_SGI            = "eth0";
        PGW_IPV4_ADDRESS_FOR_SGI              = "192.168.12.82/24";
        PGW_MASQUERADE_SGI                    = "yes";
    };
...
   # DNS address communicated to UEs
    DEFAULT_DNS_IPV4_ADDRESS     = "192.168.106.12";
    DEFAULT_DNS_SEC_IPV4_ADDRESS = "192.168.12.100";
...
}
```

HSS freediameter configuration (/usr/local/etc/oai/freeDiameter/hss_fd.conf):

```
Identity = "hss.openair4G.eur";
Realm = "openair4G.eur";
```

MME freediameter configuration (/usr/local/etc/oai/freeDiameter/mme_fd.conf):

```
Identity = "example.openair4G.eur";
Realm = "openair4G.eur";
ConnectPeer= "hss.openair4G.eur" { ConnectTo = "127.0.33.1";
No_SCTP ; No_IPv6; Prefer_TCP; No_TLS; port = 3868;
realm = "openair4G.eur";};
```

Importantly, "example" on line 1 is meant to be replaced by the hostname/FQDN choosen during the installation process explained in Appendix A.

hss configuration (/usr/local/etc/oai/hss.conf):

```
MYSQL_user   = "root";
MYSQL_pass   = "linux";
OPERATOR_key = "1006020f0a478bf6b699f15c062e42b3";
```

Here, the MYSQL password must be provided. As explained in appendix A, the easiest way is to use "linux", keeping the original HSS configuration file.

B.1 T3470 Timer

The value of the T3470 Timer is specified in the mme.conf. In order to change the value, open the mme.conf file located in "/usr/local/etc/oai":

```
# T3470 start: IDENTITY REQUEST sent
# T3470 stop: IDENTITY RESPONSE received
T3470   =  6
```

To change the timer from the standard value of 6 seconds, change the number on the last line from 6 to the desired value.

Appendix C
USRP B200mini Series Architecture

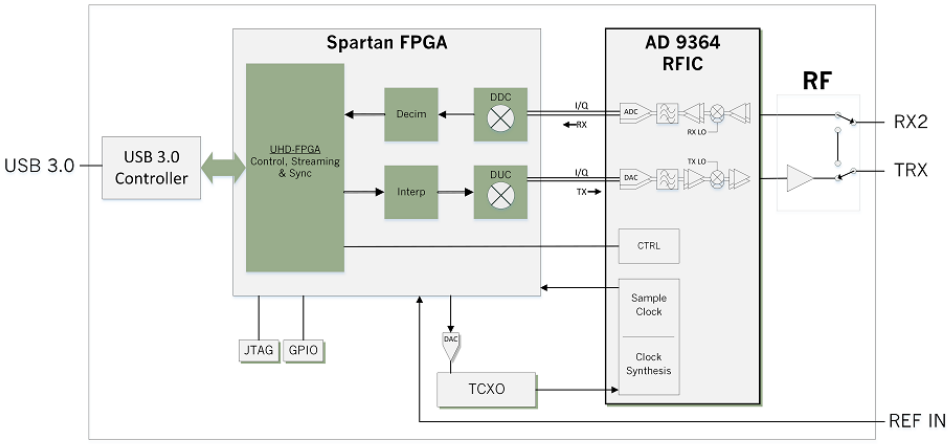


Figure C.1: USRP B200mini Series Architecture from [39]