NTNU
Norwegian University of
Science and Technology

Location Disclosure in LTE Networks by using IMSI Catcher

Christian Sørseth

Master of Telematics - Communication Networks and Networked Services
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:	Location Disclosure in LTE Networks by using IMSI Catcher
Student:	Christian Sørseth

Problem description:

An IMSI Catcher is a device that acts as a false base station to implement a man-in-the-middle attack in mobile networks. On top of disclosing the IMSI (International Mobile Subscriber Identity) and intercept network traffic, the IMSI Catchers track the movement of mobile users. Recently, low-cost IMSI Catchers were proved feasible for LTE too. LTE location attacks allow an adversary to track the presence or absence of an IMSI in a given area, sometimes even localizing the IMSI in an area tighter than a cell range.

The master thesis will investigate and analyze potential passive and active location disclosure attacks in LTE networks using IMSI Catchers. The student will build an LTE IMSI Catcher based on the open-source platform OpenAirInterface, with the main goal to collect IMSIs. The student should also analyze the possibility of collecting IMSIs passively, for example by listening and decoding broadcast paging messages sent by commercial base stations. Existing location disclosure attacks will be technically explained and analyzed and if time permits, improvements and countermeasure proposals should be considered.

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

Abstract

Long-Term Evolution (LTE) is currently being deployed in vast areas of the world and is the latest implemented standard in mobile communication. The standard is considered to have significant improvements compared to its predecessors; however, several weaknesses exists. One of the deficiencies in LTE is that a big portion of the signaling messages is transmitted without protection. International Mobile Subscriber Identity (IMSI) Catchers and Paging Catchers exploit this weakness to perform several attacks against privacy in LTE, which disrupts the communication service and weakens the credibility of mobile operators.

An IMSI Catcher is essentially a device masquerading itself as commercial Base Station (BS) used to track devices and break subscriber privacy. In this thesis, IMSI Catchers in LTE networks are studied. An LTE IMSI Catcher has been implemented using a Universal Software Radio Peripheral (USRP) and the open source platform OpenAirInterface. By the help of IMSI Catchers, an attack against subscriber privacy was conducted. The attack efficiently acquires subscription identities (IMSIs) within a limited area and then redirects subscribers back to the commercial network. The attack has been carefully tested and successfully proven feasible. It was found that the IMSI acquisition process is very efficient, and several IMSIs were collected within a few seconds of operation.

Additionally, Paging Catchers are studied in this thesis. A Paging Catcher is a tracking device used to perform attacks against subscriber privacy passively; however, unlike the IMSI Catcher, the Paging Catcher masquerades itself as a commercial User Equipment (UE). A Paging Catcher has been implemented using a USRP and the open source platform srsLTE. This thesis verifies that a Paging Catcher attack locates LTE devices within a limited area and breaks subscriber privacy. The attack illustrates that the Paging Catcher conveniently receives paging messages broadcasted by nearby BSs. The paging messages contain Temporary Mobile Subscriber Identities (TMSIs) which is mapped to social identities. The attack has successfully been proven feasible; however, the Paging Catcher is dependant of the smart paging feature to locate the subscriber precisely.

Sammendrag

Long-Term Evolution (LTE) blir i disse dager utplassert i store deler av verden og er den nyeste distribuerte standarden innen trådløs mobil kommunikasjon. Standarden anses å ha store forbedringer sammenlignet med tidligere standarder, men flere sikkerhetshull har blitt påvist. En av svakhetene til LTE er at en stor del av 'signaling' meldingene blir prosessert uten kryptering. International Mobile Subscriber Identity (IMSI) fangere kan utnytte denne svakheten til å gjennomføre flere angrep mot personvern i LTE, noe som forstyrrer kommunikasjonstjenesten og svekker troverdigheten til mobiloperatører.

En IMSI fanger er i hovedsak en enhet som utgir seg for å være en kommersiell Base Station (BS), som brukes til å spore LTE enheter og bryter personvern for abonnenter. Denne oppgaven tar for seg IMSI fangere i LTE nettverk. En IMSI fanger har blitt implementert ved hjelp av en Universal Software Radio Peripheral (USRP) og programvaren OpenAirInterface. Et angrep mot personvern for abonnenter ble gjennomført med hjelp av en IMSI fanger. Angrepet samlet effektivt abonnent identiteter (IMSIer) innenfor et begrenset område, deretter omdirigeres abonnentene tilbake til det kommersielle nettverket. Angrepet ble nøye testet og var vellykket utført. Det viste seg at IMSI fangeren var svært effektiv, og flere IMSIer ble fanget etter få sekunder.

Denne oppgaven har også studert bruken av paging fangere i LTE nettverk. En paging fanger er en sporingsenhet som brukes til å utføre angrep mot personvern for LTE abonnenter. I motsetning til IMSI fangeren, utgir paging fangeren seg for å være en kommersiell User Equipment (UE). En paging fanger har blitt implementer ved hjelp av en USRP og programvaren srsLTE. Denne oppgaven verifiserer at et paging angrep kan lokalisere LTE-enheter innenfor et begrenset område og bryter abonnentens personvern. Angrepet illustrerte at en paging fanger enkelt mottar paging meldinger fra nærliggende BSs. Paging meldingene inneholder den midlertidige identiteten til abonnenter (TMSI), som kobles til personlige identiteter. Angrepet har blitt vellykket utført, men paging fangeren er avhengig av 'smart paging' funksjonen for å lokalisere abonnenten nøyaktig.

Preface

This Master's thesis is the result of the work in Information Security in the final semester of my Master of Science degree in Telematics at Norwegian University of Science and Technology. The thesis is written under the supervision of Professor Stig Frode Mjølsnes and Ruxandra-Florentina Olimid from Department of Information Security and Communication Technology.

I would like to thank Professor Stig Frode Mjølsnes and Ruxandra-Florentina Olimid for much valuable guidance and feedback during the work with this thesis.

I would also like to thank my fellow student Christoffer Evjen Ottesen for participating in the IMSI Catcher experiment.

Trondheim, June 2017

Christian Sørseth

Contents

List of Figures

List of Tables

List of Acronyms

3GPP 3rd Generation Partnership Project.

4G 4th Generation.

AK Anonymity Key.

AKA Authentication and Key Agreement.

AMF Authentication Management Field.

AS Access Stratum.

ASN.1 Abstract Syntax Notation One.

AuC Authentication Center.

AV Authentication Vector.

BCCH Broadcast Control Channel.

BCH Broadcast Channel.

BIOS Basic Input-Output System.

BS Base Station.

CK Cipher Key.

CPU Central Processing Unit.

CRNTI Cell RNTI.

CS Circuit Switched.

DL Downlink.

DL-SCH Downlink Shared Channel.

DoS Denial-of-Service.

EARFCN E-UTRA Absolute Radio Frequency Channel Number.

EMM EPS Mobility Management.

eNodeB Evolved Node B.

EPC Evolved Packet Core.

EPS Evolved Packet System.

ETWS Earthquake and Tsunami Warning System.

E-UTRAN Evolved Universal Terrestrial Radio Access Network.

GPS Global Positioning System.

GSM Global System for Mobile Communications.

GTP GPRS Tunneling Protocol.

GUMMEI Globally Unique MME Identifier.

GUTI Globally Unique Temporary UE Identity.

HSS Home Subscriber Server.

IK Integrity Key.

IMEI International Mobile Equipment Identity.

IMS IP Multimedia Subsystem.

IMSI International Mobile Subscriber Identity.

IMT International Mobile Telecommunications.

IP Internet Protocol.

IPsec Internet Protocol Security.

ITU International Telecommunication Union.

LTE Long-Term Evolution.

MAC Message Authentication Code.

MCC Mobile Country Code.

ME Mobile Equipment.

MIB Master Information Block.

MITM Man-in-the-Middle.

MME Mobility Management Entity.

MNC Mobile Network Code.

MSIN Mobile Subscriber Identification Number.

MSISDN Mobile Station International Subscriber Directory Number.

M-TMSI MME Temporary Mobile Subscriber Identity.

NAS Non Access Stratum.

NTNU Norwegian University of Science and Technology.

OAI OpenAirInetrface.

OSI Open Systems Interconnection.

PBCH Physical Broadcast Channel.

PCCH Paging Control Channel.

PCFICH Physical Control Format Indicator Channel.

PCH Paging Channel.

PCRF Policy and Charging Rules Function.

PDCCH Physical Downlink Control Channel.

PDCP Packet Data Convergence Protocol.

PDN Packet Data Network.

PDSCH Physical Downlink Shared Channel.

PDU Protocol Data Unit.

P-GW PDN Gateway.

PHICH Physical Hybrid ARQ Indicator Channel.

PLMN Public Land Mobile Network.

PMCH Physical Multicast Channel.

PRACH Physical Random Access Channel.

P-RNTI Paging RNTI.

PS Packet Switched.

P-TMSI Packet-Temporary Mobile Subscriber Identity.

PUCCH Physical Uplink Control Channel.

PUSCH Physical Uplink Shared Channel.

QoS Quality of Service.

RAN Radio Access Network.

RES Response.

RLC Radio Link Control.

RLF Radio Link Failure.

RNTI Radio Network Temporary Identifier.

RRC Radio Resource Control.

RSRP Reference Signal Received Power.

RSRQ Reference Signal Received Quality.

S1AP S1 Application Protocol.

SCTP Stream Control Transmission Protocol.

SDR Software Defined Radio.

SFN System Frame Number.

S-GW Serving Gateway.

SI System Information.

SIB System Information Block.

SIM Subscriber Identity Module.

SI-RNTI System Information RNTI.

SMS Short Message Service.

SN ID Serving Network ID.

SPGW Serving Gateway/PDN Gateway.

SQN Sequence Number.

SRB Signalling Radio Bearer.

SRS Software Radio Systems.

S-TMSI SAE-Temporary Mobile Subscriber Identity.

TA Tracking Area.

TAC Tracking Area Code.

TAI Tracking Area Identity.

TAU Tracking Area Update.

TMSI Temporary Mobile Subscriber Identity.

UE User Equipment.

UICC Universal Integrated Circuit Card.

UL Uplink.

UMTS Universal Mobile Telecommunications System.

UP User Plane.

USB Universal Serial Bus.

USIM Universal Subscriber Identity Module.

USRP Universal Software Radio Peripheral.

XMAC Expected MAC.

XML Extensible Markup Language.

XRES Expected Response.

Chapter 1
Introduction

1.1 Motivation

Mobile communication plays a central role for most people in today's society. LTE is currently being deployed in vast areas of the world and is the latest implemented standard in mobile communication. LTE is considered to have significant improvements compared to its predecessors; in addition to the high data throughput, the security and privacy for subscribers have improved substantially. Historically, Global System for Mobile Communications (GSM) and Universal Mobile Telecommunications System (UMTS) have been the leading technology in mobile communication, after smartphones entered the marked, the demand for mobile data increased drastically. Consequently, LTE has managed to meet the growing need for mobile data and have become the leading technology in wireless mobile communication. Although the LTE security has improved compared to its predecessors, several weaknesses exist. One of the deficiencies in LTE is that a big portion of the signaling messages may be processed without protection.

An IMSI Catchers is essentially a device masquerading itself as commercial BS used to implement a Man-in-the-Middle (MITM) attack in mobile networks. In addition to disclosing the IMSI and intercept network traffic, the IMSI Catchers track the movement of mobile users. Since the IMSI Catcher is configured similarly as a commercial BS, UEs will not be able to distinguish false BSs from commercial BSs. Consequently, UEs automatically attaches to the false BS, which allows attackers to implement several attacks.

Unlike an IMSI Catcher, a Paging Catcher is used to masquerade itself as a commercial UE. The Paging Catcher collects and decodes paging messages broadcasted by commercial BSs. By definition, paging messages cannot be protected by authentication and are, therefore, a weak spot. Consequently, a Paging Catcher conveniently catches paging messages and extracts Globally Unique Temporary UE Identities (GUTIs) used to track the movement of subscribers.

1.2 Scope and Objectives

A Universal Software Radio Peripheral (USRP) and the open source software OpenAirInterface were acquired to experiment with location disclosure attacks in LTE networks. Initially, the primary goal of the thesis was to build an LTE IMSI Catcher based on OpenAirInterface and catch IMSIs. Additionally, a Paging Catcher was built, with the goal to passively catch and decode broadcast paging messages sent by commercial BSs. Paging messages were analyzed and used in attacks against privacy in LTE. Existing location disclosure attacks have been technically analyzed, and improvements have been proposed. During the process, it was decided to extend the thesis by including an experiment regarding Globally Unique Temporary UE Identity (GUTI) persistence, to determine how often a Norwegian mobile operator changes the GUTI for its subscribers.

OpenAirInterface proved to be very sensitive to version numbers and hardware models, which caused the installation and configuration to be more time-consuming than planned. Also, OpenAirInterface has not implemented the paging procedure yet. Consequently, srsLTE was chosen for the Paging Catcher. SrsLTE is an open source software similar to OpenAirInterface; however, srsLTE has fewer dependencies and is easier to install.

All prerequisites and dependencies required for installing OpenAirInterface and srsLTE are appended. Commands and configuration parameters for the IMSI Catcher and the Paging Catcher are given in a tutorial-like manner, which makes subsequent recreations convenient.

1.2.1 Objectives

The overall focus area of this thesis is the usage of IMSI Catchers in LTE networks. The goal of this thesis is to provide a thorough technical description of IMSI Catchers and how they can be used to disclose the location of subscribers in LTE networks. This thesis also discusses Paging Catchers and how they exploit unprotected paging messages to determine the location of a subscriber, in addition to a general overview of the main aspects of the LTE technology. Moreover, this thesis is divided into seven primary objectives:

1. Study the feasibility of location attacks in LTE using low-cost hardware and open source software
2. Capture IMSIs or other sensitive information of subscribers
3. Obtain the required measurement configurations for the IMSI Catcher by sniffing System Information Block (SIB) messages from the target cell

4. Build and set up an IMSI Catcher based on the open source platform OpenAir-Interface and a USRP, subsequently use the IMSI Catcher to disclose IMSIs and track the movement of subscribers
5. Build and set up a Paging Catcher based on the open source platform srsLTE and a USRP, subsequently use the Paging Catcher to create a mapping between GUTI and social identity mapping, and track the movement of subscribers
6. Use the Paging Catcher to check GUTI persistence for one of the Norwegian operators
7. Analyze existing location disclosure attacks and propose improvements

1.3 Work Method

The research methodology adopted in this thesis is divided into four main segments.

The first segment was exclusively a literature study which consisted of analyzing previous related work. The study was primarily based on the papers described in Section 1.5 and the LTE standardization provided by 3rd Generation Partnership Project (3GPP).

The second segment concerned practical experiments. Different configurations and software were tested in diverse scenarios. An LTE IMSI Catcher was built using OpenAirInterface and a USRP. Additionally, a Paging Catcher has been constructed using srsLTE and a USRP.

The third segment consisted of analyzing the data collected in the second phase. The data gathered by the IMSI Cather and the Paging Catcher were technically analyzed and used to break subscriber privacy.

The fourth segment concerned an analysis of theoretical premises.

1.4 Contributions

This thesis provides a theoretical and practical study of LTE IMSI Catchers and Paging Catchers. The primary contribution of this thesis is the implementation of an LTE IMSI Catcher, and how it is used to disclose the location of a subscriber. Chapter 3 describe how an LTE IMSI Catcher exploits features in the LTE specification to steal IMSIs and subsequently reconnects the subscribers to the commercial LTE network. A functional description on how to configure the IMSI Catcher, as well as a technical description of the functionality are given in Chapter 3. Appendix A and Appendix B provides a functional description of how to build and configure the IMSI Catcher.

Additionally, methods for obtaining subscriber identities passively are proposed. Chapter 4 describes how to implement a Paging Catcher able to acquire and decode broadcast paging messages sent by commercial BSs. Paging messages contain subscriber identities and were exploited in attacks against privacy in LTE. The Paging Catcher were also used to analyze the GUTI persistence for a Norwegian mobile operator. Chapter 4 also describes how to catch SIB messages from commercial BSs passively. SIB messages contain detailed information about the mobile operator and are utilized to configure the IMSI Catcher and the Paging Catcher.

Chapter 5 includes improvements for existing location disclosure attacks. The improvements simplify the existing attacks and enrich the overall outcome.

1.5 Related Work

1.5.1 Related Work in LTE

Mjølsnes and Olimid published a simplified LTE IMSI Catcher during the work with this thesis [MO17]. They implemented the IMSI Catcher using a USRP and OpenAirInterface, and the results were very similar to the results obtained in this thesis. However, their proposal denies the subscribers access to the commercial network after obtaining the IMSI, whereas the IMSI Catcher proposed in this thesis, redirects the subscriber back to the commercial network after obtaining the IMSI. Unlike the IMSI Catcher proposed by Mjølsnes and Olimid, the IMSI Catcher in this thesis collects the IMSI with a low probability of the subscriber noticing it.

Shaik et al. presented in 2016 a highly relevant research regarding privacy in LTE networks [SBA+15]. They performed attacks able to accurately locate subscribers within a given area by using a USRP and the open source platform srsLTE [Sof]. Moreover, their attacks are discussed in Chapter 5.

Rupprecht et al. developed in 2016 a framework for identifying implementation flaws in LTE by using OpenAirInterface [RJP16]. Their research discovered several security flaws in the LTE implementation. Moreover, their results were not directly related to location disclosure and movement tracking; however, the framework and test environment were highly relevant to this thesis.

1.5.2 Related Work in Previous Generations Systems

Considerable research has been implemented on IMSI Catcher in GSM and UMTS. Retterstøl showed in 2015 an efficient implementation of an IMSI Catcher operating in GSM [Ret15]. He built an IMSI Catcher based on a USRP and the open source platform OpenBTS [Opeb].

Ooi presented in 2015 a general overview of functionality and capabilities of IMSI Catchers [Ooi15]. He also presented several countermeasures against IMSI Catchers and proposals on how to distinguish IMSI Catchers from legitimate BSs.

Meyer et al. presented in 2004 an attack that allows an adversary to impersonate a GSM BS to a UMTS subscriber, regardless whether UMTS authentication is used. [Mey04]. They showed that the attack could be used to eavesdrop on all traffic initiated by the subscriber.

1.6 Outline

This thesis is divided into five chapters, excluding this introduction chapter:

Chapter 2 includes the fundamental concepts of LTE that are relevant to get a better understanding of the content of this thesis. This chapter also includes the security architecture of LTE and known vulnerabilities.

Chapter 3 includes a technical description of IMSI Catchers in LTE and how to use them for location disclosure and movement tracking.

Chapter 4 includes a technical description of Paging Catchers and how they passively sniff paging messages broadcasted by commercial Evolved Node Bs (eNodeBs). This chapter also analyzes GUTI persistence for a Norwegian operator.

Chapter 5 includes existing location disclosure attacks in LTE networks, in addition to improvement proposals.

Chapter 6 concludes the work done in this thesis and potential further work.

Chapter 2
LTE

This chapter provides a general overview of the fundamental parts of the LTE technology that are necessary to understand the content of this thesis. The chapter contains an introduction to the LTE architecture and explains how various network components interact. Also, it explains how the protocol architecture is divided into planes and how different channels are used to transport data across the LTE radio interface. Lastly, it includes an overview of the Public Land Mobile Networks (PLMNs) in Norway and the security aspects in LTE.

2.1 Overview

LTE, also known as 4th Generation (4G)[1], is the latest standard in mobile network technology and is supported by most smartphones. 3GPP developed the standard with the aim to increase downlink and uplink peak data rates, create scalable carrier bandwidths, and make a purely Internet Protocol (IP) based network architecture [3GP]. In addition to the significant functionality improvements, its security and privacy have also improved a lot compared to its predecessors. As of today, LTE is the fastest developing mobile network technology of all time and are commercially launched in more than 70% of the world [GSA15]. Furthermore, the 5th Generation (5G) technology is under development, and 3GPP estimates to deploy the standard in 2020 [Gio16].

2.2 LTE Network Architecture

2.2.1 Overview

The LTE network architecture is roughly divided into three parts: the access part called the Evolved Universal Terrestrial Radio Access Network (E-UTRAN), the

[1] LTE does not fulfill the 4G requirements stated by International Mobile Telecommunications (IMT); however, they have eventually agreed to name it 4G [Pro].

core part called the Evolved Packet Core (EPC), and the UE. Furthermore, the E-UTRAN and EPC are divided into several network components, each playing an important role in the complete LTE network architecture. Figure 2.1 illustrates the complete overview of the LTE network architecture, showing the relationship between UE, E-UTRAN, EPC and their corresponding network components. This section will describe all the relevant LTE components and explain their role in the network.

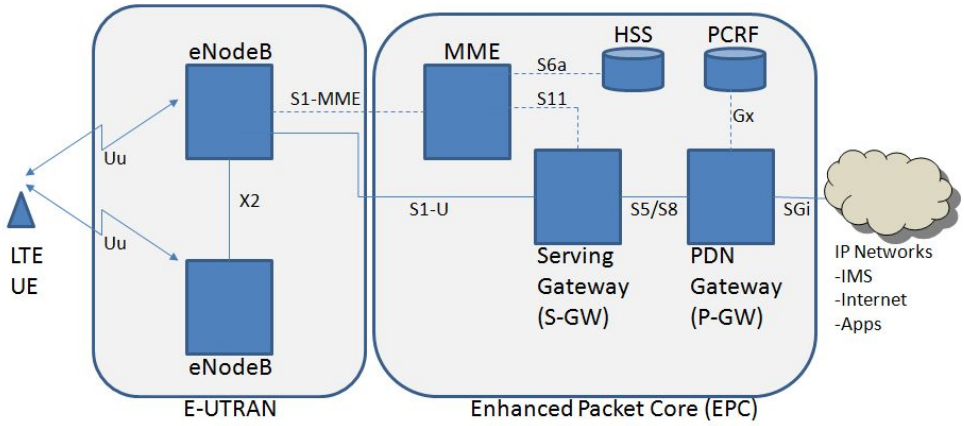

Figure 2.1: LTE network architecture. Source: [New]

2.2.2 Evolved Packet Core (EPC)

As illustrated in Figure 2.1 the EPC consists of Mobility Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (S-GW), PDN Gateway (P-GW), and Policy and Charging Rules Function (PCRF). The PCRF will not be discussed as it is not relevant to this thesis.

Mobility Management Entity (MME)

The MME is a key control plane entity within the EPC, providing an interface towards the E-UTRAN. The primary responsibility of the MME is to manage the accessibility of network connections, allocate network resources, and authenticate UEs [Sri12]. A single MME is managing the connection to multiple eNodeBs; however, to handle the massive signaling load in mobile networks, MMEs can be grouped together in a pool [Sri12]. Section 2.5.3 describe the MME pool in detail. The authentication procedure is the initial step performed when a UE first connect to a network, and the MME has the overall responsibility for this procedure. Section 2.6.3 provides a detailed description of the LTE authentication and key agreement process.

Home Subscriber Server (HSS)

The HSS is essentially a database containing user-related and subscriber-related information such as Globally Unique MME Identifier (GUMMEI), IMSI, authentication key K, Quality of Service (QoS) profile, and roaming restrictions [Fre17, Luc09]. Moreover, the HSS plays a central role in the authentication and key agreement process, where it has the overall responsibility to decide if a UE may access an LTE network.

Commonly, the HSS integrates the Authentication Center (AuC), responsible for generating security keys and authentication vectors used in the authentication and key agreement process [SBT11].

Packet Data Network Gateway (P-GW)

P-GW is the exit and entry node for UE traffic destined for external packet data networks, such as IP Multimedia Subsystem (IMS) and the Internet. The primary responsibilities of the P-GW are to perform QoS provisioning by means of deciding who can access which resources in the network (policy enforcement) [Pro]. The P-GW is also responsible for allocating IP addresses, packet filtering, and flow-based charging for each UE [Luc09]. Since the P-GW is the interconnection node between the EPC and external Packet Data Networks (PDNs), is it acting as a mobility anchor for communication with non-3GGP technologies [SBT11].

Serving Gateway (S-GW)

S-GW is the interconnection node between the EPC and the E-UTRAN. The S-GW routes and forwards incoming and outgoing IP packets to/from the UE [Pro]. It also acts as a mobility anchor for intra-LTE mobility, meaning that the same S-GW is used during handover to eNodeBs located in different Tracking Areas (TAs) [3GP08d]. Also, the S-GW is responsible for initiating paging when the UE is in *IDLE* mode [Pro].

2.2.3 Evolved Universal Terrestrial Radio Access Network (E-UTRAN)

As illustrated in Figure 2.1, the access network (E-UTRAN) consists of several eNodeBs. Moreover, the same MME may connect several eNodeBs in the same E-UTRAN. In E-UTRAN there is no designated controller for regular user traffic, and the architecture is said to be flat [SBT11].

Evolved Node B (eNodeB)

The eNodeB is known to most people as a BS and is responsible for all LTE-related radio functionality. Each eNodeB is in charge of serving a coverage area, which is divided into several sectors known as cells [SBA+15]. Below is a list of the most relevant eNodeB functionality [3GP10]:

- **Scheduling and Transmission** Helps MME transport signaling messages and broadcast information to UEs. It is also responsible for routing user plane data from the UE to the S-GW
- **MME Selection** The eNodeB is responsible for selecting a valid MME during the attach procedure
- **Compression** Performs IP header compression for better utilization of the radio interface.
- **Security** Applies encryption to user data sent over the radio interface. Details on how security is handled in LTE are further described in Section 2.6

All of the above-listed functions reside in the eNodeB. By performing the operations in the access network instead of the core network makes the interaction between protocol layers much more efficient. Consequently, latency is reduced and efficiency is improved for the radio access network. Also, by distributing the control to each eNodeB the need for one centralized controller is avoided, resulting in a potential cost saving and bypassing single points of failure.

2.2.4 User Equipment (UE)

The UE is recognized by most people as a cell phone, being an endpoint for cellular traffic. As defined in the standards, UE consists of Mobile Equipment (ME), Universal Integrated Circuit Card (UICC), also known as Subscriber Identity Module (SIM) and a Java application, known as Universal Subscriber Identity Module (USIM) [Cic16]. The serving network provides the UE with access to the services offered by the home network. One of the most important modules of the UE is the UICC, a multi-application platform including applications such as:

- **USIM (Universal Subscriber Identity Module)** Application required in the Authentication and Key Agreement (AKA) procedure, which stores important information such as security keys, Mobile Station International Subscriber Directory Number (MSISDN), Mobile Network Code (MNC), and Mobile Country Code (MCC) [Cic16]

– **SIM (Subscriber Identity Module)** Application used to communicate with GSM systems

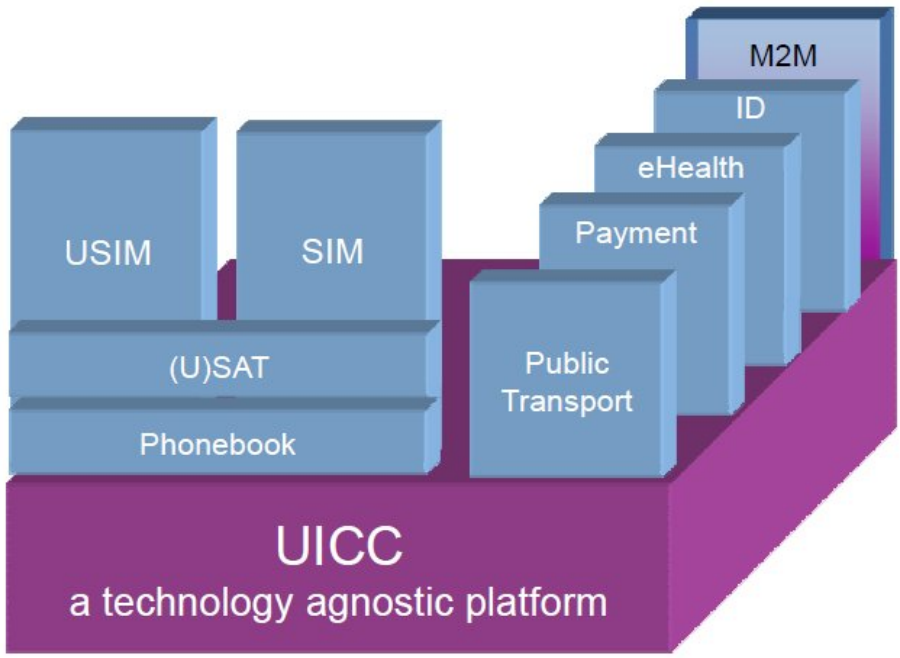

Figure 2.2: UICC architecture providing a clear separation of the applications residing on it. Source: [Zah12].

SIM is the predecessor to the UICC, initially designed to operate in the GSM network and could only host one application. As UMTS and LTE emerged the SIM card was replaced by the UICC, currently used today. Figure 2.2 illustrates the structure of the UICC and the clear separation of the applications residing on it. In addition to being a multi-application platform is it IP-connected, enabling subscribers to access cloud-based services and applications such as mobile banking [SIM11]. UICC is the only subscriber-owned component in an LTE network that an operator has no physical control over. However, due to the UICC always being IP-connected, operators can remotely do changes to the applications residing on the UICC, for example, changing the roaming agreements.

2.3 Protocol Architecture

EPC systems use multiple protocols for the communication between the UE and the eNodeB. Each protocol performs operations on the user plane and/or the control plane. The user plane is used to route user data between the UE and the MME, while the control plane is used to carry signaling messages between the UE and the S-GW [Cic16].

2.3.1 User Plane

IP packets destined for a UE are encapsulated in an EPC-specific tunneling protocol and transported from the P-GW through the S-GW to the eNodeB, where the packet is transmitted to the UE over the air. The user data is encapsulated in the GPRS Tunneling Protocol (GTP) during the transportation from the P-GW to the eNodeB. The E-UTRAN user plane protocol stack is marked blue in Figure 2.3 [Luc09]. The protocol stack is composed of the Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC) and Medium Access Control (MAC) [3GP10].

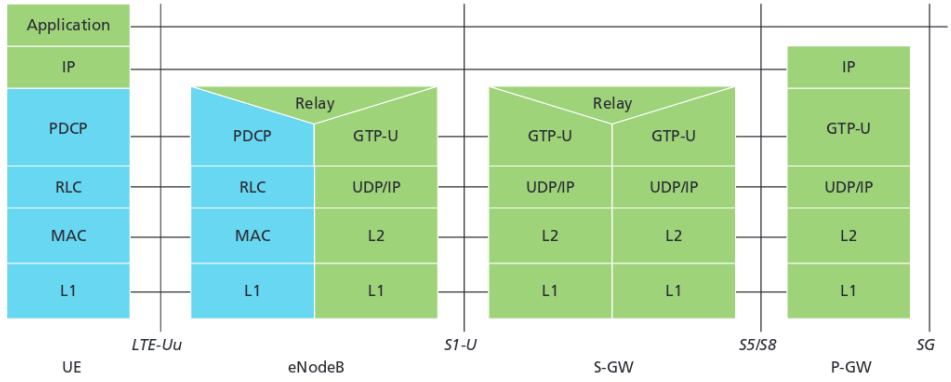

Figure 2.3: User plane protocol stack. Source: [Luc09].

2.3.2 Control Plane

The control plane includes functionality such as paging, broadcasting system information, UE measurement reporting, authentication, and EPC bearer management [3GP10]. Figure 2.4 illustrates the protocol stack for the control plane between the UE and the eNodeB. Non Access Stratum (NAS) is the network layer communication between the UE and the MME, while the blue part of Figure 2.4 depicts the Access Stratum (AS) protocols used for communication between the UE and the eNodeB. The control plane contains the same protocols as in the user plane protocol

stack; additionally, the control plane includes the Radio Resource Control (RRC) protocol. PDCP, RLC, and MAC have the same functionality as for the user plane; except that PDCP does not perform header compression in the control plane [Luc09]. Section 2.3.3 explains all the protocols used in the user plane and the control plane.

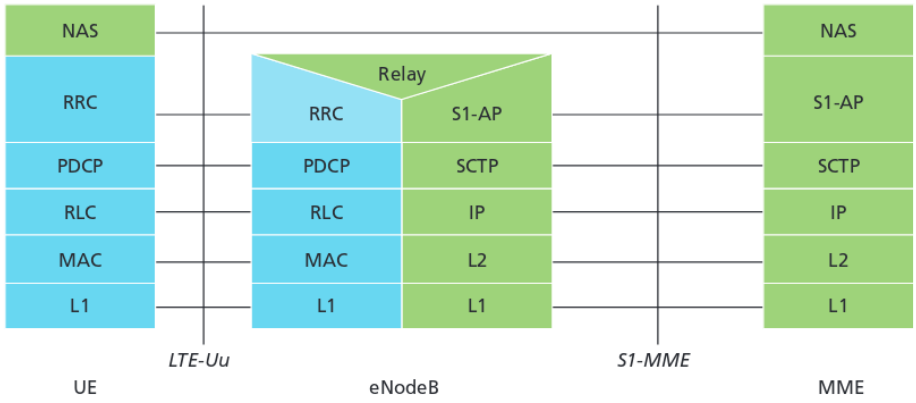

Figure 2.4: Control plane protocol stack. Source: [Luc09].

2.3.3 User Plane and Control Plane Protocols

Non Access Stratum (NAS)

NAS signaling is responsible for generation and allocation of unique temporary identities called SAE-Temporary Mobile Subscriber Identity (S-TMSI), which are used by the MME to identify the UE [SBT11]. The S-TMSI is only temporary and should regularly be changed by the network operator to maintain the privacy of subscribers. Also, NAS signaling can be used to check whether a UE is authorized to camp on the service providers PLMN, and have the ability to enforce UE roaming restriction.

Radio Resource Control (RRC)

RRC is the link signaling protocol for the AS. As illustrated in Figure 2.4 RRC tasks are performed in the eNodeB to maintain a flat structure. Some of the RRC sublayer functions include broadcasting system information, paging, allocating temporary identities between the eNodeB and the UE, key management, and UE measurement reporting [3GP10].

Packet Data Convergence Protocol (PDCP)

The primary function of the PDCP is to carry RRC signaling and user data, in addition to handle ciphering, deciphering and integrity protection [3GP10]. The PDCP layer also includes the functionality of user plane header compression and encryption. The compression procedure consists of replacing the IP header by a token of 3-4 bytes, minimizing the amount of header data that is sent over the air [LLM+09].

Radio Link Control (RLC)

RLC is a sublayer residing between the PDCP and the MAC layer. The primary functions of the RLC are to transfer upper layer Protocol Data Units (PDUs). In addition to concatenation, segmentation, and reassembly of data that has been passed down from a higher sublayer [3GP10].

Medium Access Control (MAC)

The Message Authentication Code (MAC) sublayer has the responsibility to create a mapping between transport channels and logical channels, in addition to deciding which transport format to use. Subsequently, MAC is responsible for selecting the prioritized logical channel for a particular UE and differentiating between UEs using dynamic scheduling [3GP10].

S1 Application Protocol (S1AP) and Stream Control Transmission Protocol (SCTP)

S1 Application Protocol (S1AP) handles signaling and paging between the E-UTRAN and the EPC. Additionally, S1AP is responsible for carrying NAS signaling functions between the MME and the UE in the control plane [3GP08b]. Stream Control Transmission Protocol (SCTP) has the responsibility to ensure a reliable delivery of signaling messages [Luc09].

Physical Layer (L1), Data Link Layer (L2) and IP Layer

the physical layer, the data link layer, and the IP layer have the same function as in the Open Systems Interconnection (OSI) reference model, specified in [Bra89].

2.4 Channel Hierarchy

2.4.1 Channel Types

To be able to transport data across the LTE radio interface, different channels are used. By dividing into different channels, data can be segregated and efficiently

carried in an orderly fashion. LTE has defined three different channel types, used to group different types of data: physical channels, transport channels, and logical channels. Figure 2.5 illustrates the LTE channel hierarchy, whereas the most relevant channel types for this thesis, are marked in red.

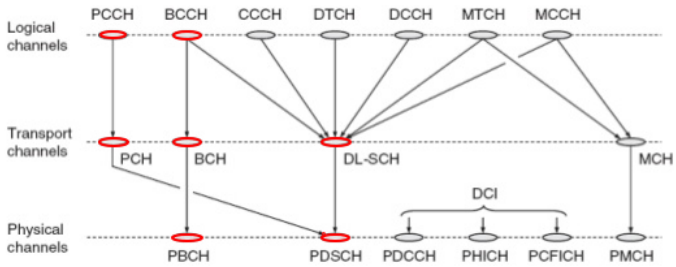

Figure 2.5: Mapping between logical, transport, and physical channels in LTE. Source: [Cho10].

2.4.2 Logical Channels

Logical channels have the overall responsibility to define the type of data transmitted over the air [3GP10]. The logical channels are mainly divided into two categories: traffic channels carrying user plane data, and control channels carrying signaling messages. The following logical channels are considered relevant for this thesis:

- **Paging Control Channel (PCCH):** A channel used to transfer paging messages and system information change notifications [3GP10]. The Paging Control Channel (PCCH) is used to carry paging messages when the network doesn't know which cell a UE might camp.

- **Broadcast Control Channel (BCCH):** A downlink channel used to broadcast system information. The Broadcast Control Channel (BCCH) is either mapped to the Broadcast Channel (BCH) or the Downlink Shared Channel (DL-SCH) dependent on the data it is transferring.

2.4.3 Transport Channels

Transport channels define how and with what characteristics data are transmitted over the air [3GP10]. Figure 2.5 depicts the mapping between the logical channels and the transport channels. The following transport channels are considered relevant for this thesis:

- **Paging Channel (PCH):** The Paging Channel (PCH) is responsible for broadcasting paging messages in the entire coverage area of the cell. The PCH channel maps to the physical channel Physical Downlink Shared Channel (PDSCH), which is dynamically allocated [3GP10].
- **Broadcast Channel (BCH):** Similarly to the PCH channel is the BCH channel responsible for broadcasting data to the entire coverage area of the cell [3GP10]. Unlike PCH, the BCH transport format is fixed and carries Master Information Blocks (MIBs) containing system information.
- **Downlink Shared Channel (DL-SCH):** DL-SCH is the primary transport channel for data transfer, and multiple logical channels map to it. In addition to transmitting application data, DL-SCH is used to broadcast SIBs and signaling messages.

2.4.4 Physical Channels

Physical channels define where data is transmitted over the air. Physical channels are used to carry data and signaling messages among the different levels of the physical layer [Tut17]. Below is a list of the most relevant physical channels for this thesis:

- **Physical Broadcast Channel (PBCH):** The Physical Broadcast Channel (PBCH) is used to transmit system information to UEs accessing a new network. The system information is carried in a MIB message and broadcasted independent of any subscribers presence [Poo12].
- **Physical Downlink Shared Channel (PDSCH):** The PDSCH is the primary channel used to transmit data over the air and is dynamically allocated to subscribers. Also, PDSCH carries broadcast messages not sent by the PBCH, which includes SIBs and paging messages [3GP08c].

2.5 LTE PLMNs in Norway

Currently, there are three PLMNs providing LTE services in Norway: Telenor, Telia, and ice.net. A PLMN is uniquely identified by a PLMN ID, which is composed of the MCC and the MNC.

2.5.1 PLMN ID Allocation in Norway

The MCC consists of a three-digit number used to identify the homeland of the mobile network operator. The MCC of Norway is 242. The MNC consists of a two or three digit number used to identify the mobile network operator uniquely. Table 2.1 shows the allocated MNCs for the leading commercial mobile operators in Norway.

Table 2.1: MCC and MNC distribution for three PLMNs in Norway [Int16].

PLMN	MNC	MCC
ice.net	14	242
Telia	02	242
Telenor	01	242

2.5.2 LTE Frequency Allocation in Norway

Norwegian PLMNs have been allocated Downlink (DL) and Uplink (UL) frequencies in four E-UTRA bands: band 3 (1800 MHz), band 7 (2600 MHz), band 20 (800 MHz), and band 31(450 MHz). Telia and Telenor have frequencies in band 3, 7, and 20 while ice.net has frequencies in band 3, 20, and 31 [Nas]. Band 3 and band 7 are common for the three PLMNs in Norway, and Table 2.2 provides a complete overview of all the allocated LTE frequencies in these bands.

Table 2.2: LTE frequency distribution in E-UTRA band 20 and band 3, as of 04.04.2017 [Nas].

	Band 20 (800MHz)		Band 3 (1800MHz)	
PLMN	DL (MHz)	UL (MHz)	DL (MHz)	UL (MHz)
ice.net	791 - 801	832 - 842	1765 - 1785	1860 - 1880
Telia	801 - 811	842 - 852	1710 - 1715 1745 - 1765	1805 - 1810 1840 - 1860
Telenor	811 - 821	852 - 862	1715 - 1745	1810 - 1840

2.5.3 Network Areas

The LTE network architecture can be divided into three areas: MME pool area, S-GW service area, and TA [Cox12]. The intention with the MME pool area is to distribute the signaling load among several MMEs and hence reduce the processing load for each MME. An MME pool area is typically covering a large geographical area such as densely populated cities [Cox12]. The S-GW service area has a similar structure as the MME pool area; however, an S-GW service area does not necessarily have to cover the same area as the MME pool area [Cox12].

MME pool areas and S-GW service areas consist of one or more TAs. A TA contains multiple BSs and is used to track the movement of UEs that are in standby mode [Cox12]. The Tracking Area Identity (TAI) uniquely identifies TAs; moreover, the TA can be identified within a particular network using the Tracking Area Code (TAC) [Cox12]. Figure 2.6 illustrates the relation between the MME pool area, the S-GW service area, and the TA.

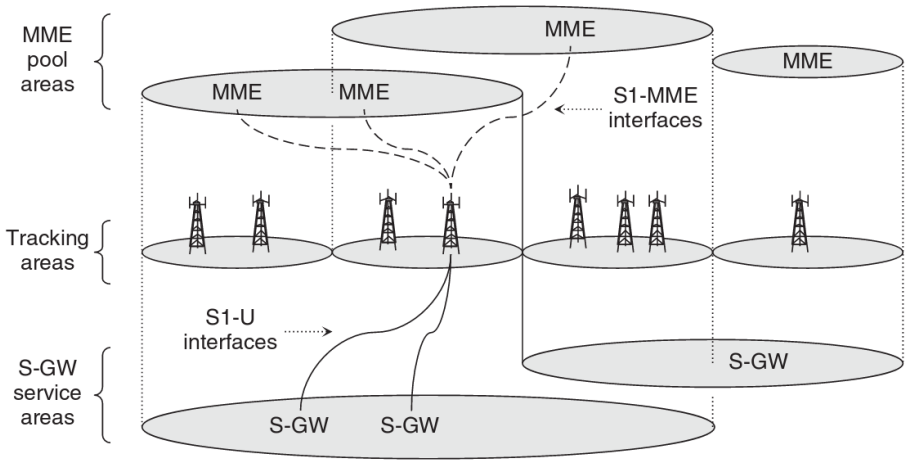

Figure 2.6: The relation between MME pool area, SGW service area, and TA. Source: [Cox12].

2.6 LTE Security

2.6.1 Overview

The security architecture of the EPC is mainly based on the UMTS architecture; however, new extensions and improvements have been implemented to increase the security of LTE. Consequently, LTE provides mutual authentication between the UE and the EPC making attacks such as MITM difficult to perform and strong encryption algorithms makes content hard to obtain. Although LTE has several solid built-in security mechanisms, sadly these are optional, and many mobile operators tend to skip them. The security mechanisms can only be activated by the mobile operators, and the subscribers have no knowledge if the parameters are activated or not. Section 2.7 describe the vulnerabilities existing in LTE networks.

In the following, a general description of the EPC security concepts is provided. Figure 2.7 illustrates how the LTE architecture implements the security features.

After the UE have connected to a network, it submits the subscriber identity to the EPC via the eNodeB; consequently, the MME queries the HSS if the UE is allowed access to the network. Additionally, the MME request the HSS for authentication data and initiates the authentication procedure if the UE identity is known. After completion of the authentication procedure, both the UE and the MME share the same master key, K_{ASME}.

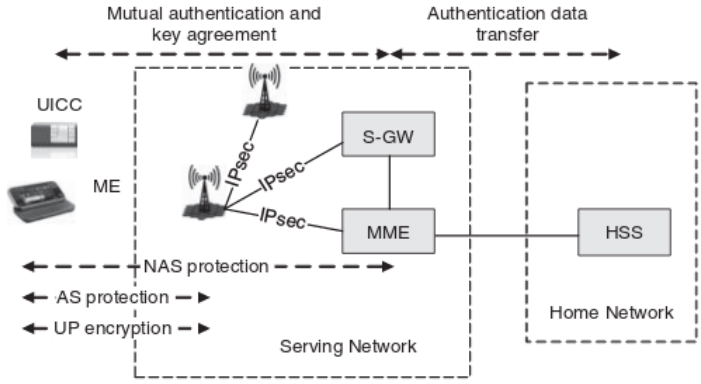

Figure 2.7: LTE security architecture. Source: [FHMN12].

Subsequently, K_{ASME} adopts further keys, used to ensure confidentiality and integrity protection of signaling messages between the MME and the UE [FHMN12]. The signaling protection is called NAS protection and is illustrated in Figure 2.7. As illustrated in Figure 2.8, K_{ASME} derives three keys: K_{NASenc}, K_{NASint}, and K_{eNB}. K_{NASenc} is used for confidentiality protection, and K_{NASint} is used for integrity protection. K_{eNB} is used to ensure User Plane (UP) confidentiality between the eNodeB and the UE, shown as UP encryption in Figure 2.7. Within the serving network, signaling and user data messages are confidentiality and integrity protected by Internet Protocol Security (IPsec).

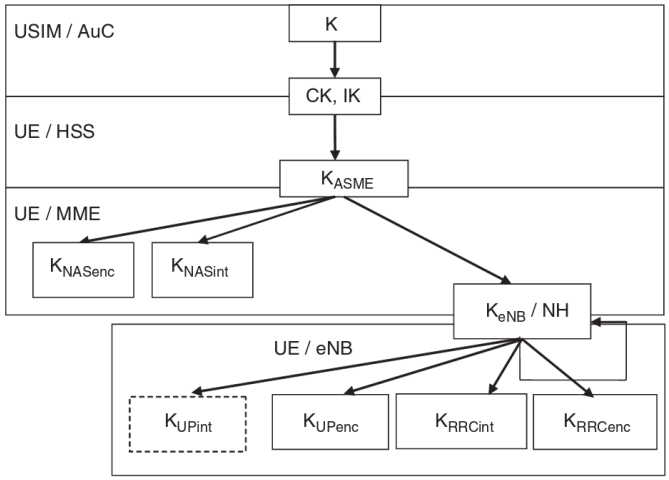

Figure 2.8: LTE key hierarchy. Source: [FHMN12].

2.6.2 Identification

Subscriber and terminal identification is a fundamental aspect of LTE systems, used by the AKA procedure to authenticate subscribers. Consequently, mechanisms to protect and uniquely allocate identities are necessary to maintain the security of LTE networks. Section 2.6.3 discuss the authentication and key agreement process further.

International Mobile Subscriber Identity (IMSI)

IMSI is a unique static identity allocated to each subscriber in an LTE system. IMSI was first introduced in the GSM standard, but the structure has remained the same for UMTS and LTE. The IMSI play a crucial role in the LTE AKA as the IMSI identifies the authentication key K. The authentication key is only stored in the AuC located in the EPC, and the USIM located in the UE. The IMSI is no more than 15 digits and is composed of MCC, MNC, and Mobile Subscriber Identification Number (MSIN) [3GP12b]. The MSIN uniquely identify a subscriber within a PLMN while the MCC and the MNC identify the country and the network operator respectively [3GP12b]. Figure 2.9 illustrates the composition of the IMSI structure.

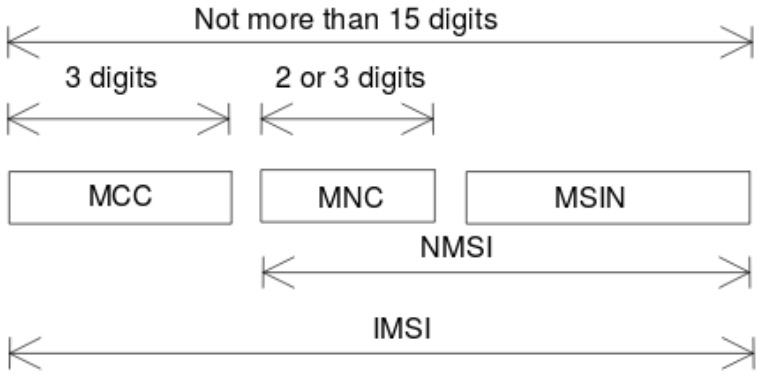

Figure 2.9: IMSI structure, composed of MCC, MNC and MSIN. Source: [3GP12b].

Globally Unique Temporary UE Identity (GUTI)

GUTI is a unique temporary identity allocated to the UE by the MME. GUTI identification is unambiguous and prevents permanent identity (IMSI) disclosure; hence mobile operators should frequently change the GUTI to maintain subscriber identity confidentiality[2]. The UE may receive a GUTI from the MME in an *Attach Accept* message or in a *Tracking Area Update Accept* message [3GP14a]. Section 3.5

[2]The GUTI update interval is operator-specific and may vary among network operators.

discuss the Tracking Area Update (TAU) procedure, and the attach procedure in further detail. The GUTI structure is divided into two parts:

- **GUMMEI** is the first part of the GUTI used to identify the MME which allocated the GUTI. GUMMEI is divided into MCC, MNC, MME Group ID, and MME Code. MME Group ID is used to identify a cluster of MMEs, and MME Code is used to identify an individual MME residing in an MME cluster. MCC and MNC have the same structure as in the IMSI.
- **MME Temporary Mobile Subscriber Identity (M-TMSI)** is the last 32 bits of the GUTI used to identify the UE temporarily. However, for paging purposes M-TMSI is replaced by S-TMSI which is constructed by the MME Code and the M-TMSI [3GP12b].

Figure 2.10 illustrates the full GUTI structure and how the different parts are composed.

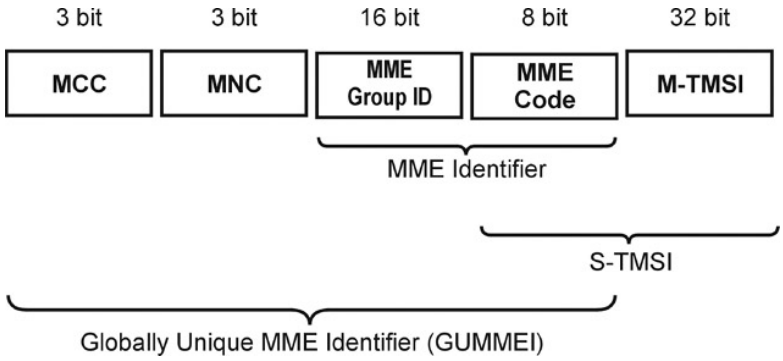

Figure 2.10: GUTI structure, composed of GUMMEI and MTMSI. Source: [KG10].

International Mobile Equipment Identity (IMEI)

International Mobile Equipment Identity (IMEI) is a unique permanent identity used by GSM, UMTS, and LTE to identify an ME. Notably, the IMEI is locked to the ME and does not change under any circumstances.

2.6.3 Authentication and Key Agreement Procedure

AKA is the authentication and key agreement procedure used by PLMNs to ensure that only authorized UEs are allowed to access their network. The AKA procedure is initiated when a UE wants to communicate with a serving network, but do not

share a security context. The procedure is illustrated in Figure 2.11 and can roughly be divided into three operations [FHMN12]:

- Initially, the UE requests to authenticate to the network by passing its IMSI or GUTI to the MME. Consequently, the MME requests authentication vectors from the HSS.
- The following step performs mutual authentication and key establishment between the UE and the serving network.
- Upon successful AKA, authentication data is exchanged between and within serving networks.

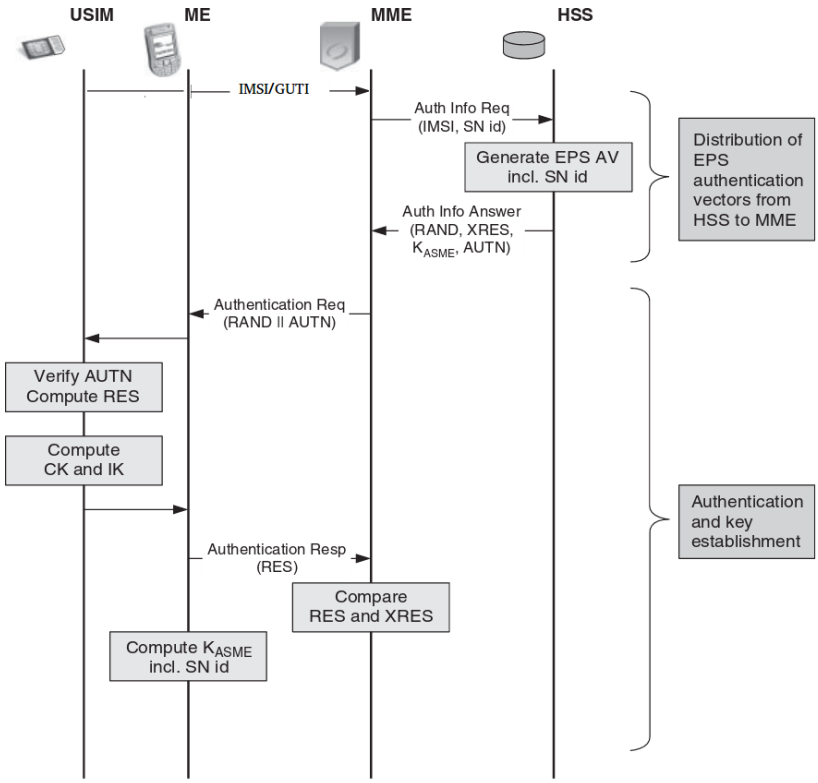

Figure 2.11: LTE authentication and key agreement (AKA) message exchange. Source: [FHMN12].

Distribution of Authentication Vectors

Distribution of authentication vectors is the first operation in the AKA process. The AKA procedure is initiated when a UE wants to connect to a network without having the required security context. The UE indicates to the serving network that it wants to connect to the network by passing its identity. Ideally only the temporary identity (GUTI) should be used in this process; however, if the serving network is unable to retrieve the IMSI from the GUTI, it invokes a user identification mechanism requesting the permanent UE ID (IMSI) [3GP12a]. Furthermore, the MME creates an *Authentication Information Request (Auth Info Req)* containing the IMSI and the Serving Network ID (SN ID). Subsequently, the HSS generates an *Authentication Information Answer (Auth Info Answer)* message and sends it back to the MME. The *Authentication Information Answer* is composed of a random number (RAND), Expected Response (XRES), master key K_{ASME}, and an authentication token (AUTN). Moreover, AUTN consists of Sequence Number (SQN), Anonymity Key (AK), Authentication Management Field (AMF), and MAC. Figure 2.12 illustrates the complete AUTN structure.

Mutual Authentication and Key Agreement

The overall objective of this part is to generate and distribute a shared local master key K_{ASME} between the UE and the MME [FHMN12]. Additionally, the serving network should authenticate the UE, and the UE should authenticate the serving network (mutual authentication). MME invokes the *Authentication Request* (*Authentication Req*) procedure containing the RAND and the AUTN. RAND is an unpredictable random number used by the UE to calculate K_{ASME}, while AUTN includes parameters used by the UE to authenticate the network. As illustrated in Figure 2.12, USIM computes the anonymity key AK and retrieves the SQN. Subsequently, USIM authenticates the serving network by verifying that SQN is in the correct range [3GP08a]. USIM also check that the calculated Expected MAC (XMAC) is equal to the received MAC, to make sure the received data is intact [Leu12]. If SQN has the correct value, the USIM replies the MME with an *Authentication Response* (*Authentication Resp*) message containing the Response (RES).

The MME compares the received RES with the expected value XRES, if the values are equal, the serving network has successfully authenticated the UE. Subsequently, the UE uses the Cipher Key (CK) and the Integrity Key (IK) to calculate the master key K_{ASME}; as a result, both the UE and the serving network have authenticated each other (mutual authentication) and successfully established security keys.

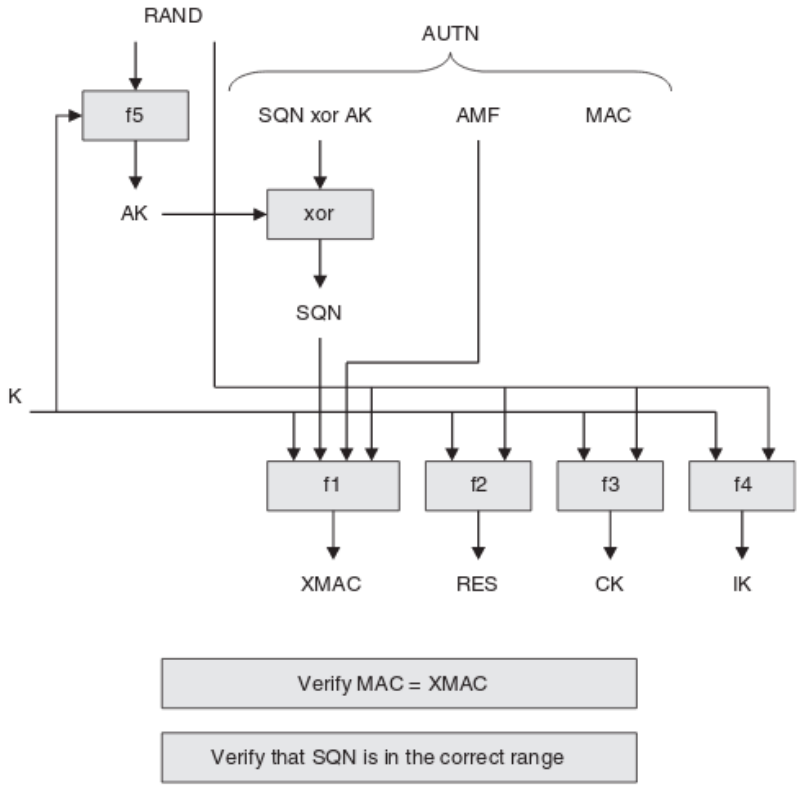

Figure 2.12: Authentication and key generation functions. Source: [3GP08a].

Exchange of Authentication Data

In general, when a UE wants to connect to an LTE network, it attaches the GUTI to a *Tracking Area Update Request* or an *Attach Request* message and passes it to the MME [3GP14a]. However, if the GUTI is unknown, the MME can either request the IMSI from the UE and break identity confidentiality, or ask the previous MME to translate the GUTI to an interpretable identity, such as the IMSI [FHMN12].

2.6.4 Difference Between GSM/UMTS and LTE Security

Cryptographic Algorithms and Cryptographic Keys

LTE systems are applying cryptographic algorithms to ensure confidentiality and integrity protection for most of the data traversing the eNodeB. The cryptographic algorithms and the usage of keys are very similar in LTE and UMTS systems [FHMN12]; however, the key hierarchy and key management are more complex in LTE. The LTE AKA procedure only generates an intermediate key K_{ASME} while

the UMTS AKA procedure uses a chain of keys. The security benefit of using an intermediate key is to ensure that each key is only functional in one particular context (cryptographic key separation) [FHMN12].

User Identity

GSM, UMTS, and LTE all have different naming conventions for their temporary identity; however, all of them are used to maintain the confidentiality of the user identity. Temporary Mobile Subscriber Identity (TMSI) temporarily identifies a subscriber in a Circuit Switched (CS) domain, while Packet-Temporary Mobile Subscriber Identity (P-TMSI) temporarily identifies a subscriber in a Packet Switched (PS) domain [NN03]. As mentioned in Section 2.6.2, GUTI is used for the services provided by the MME. Consequently, LTE-enabled devices may allocate one TMSI, one P-TMSI, and one GUTI to support GSM/UMTS handover.

2.7 Vulnerabilities in LTE

Previous research has discovered that even with mutual authentication and strong encryption algorithms, a big portion of the signaling messages is sent as plaintext. These are broadcast messages sent to all surrounding base stations (including IMSI Catchers) and can easily be sniffed by a malicious person [LJL+16]. The NAS signaling messages listed below may be processed by the EPS Mobility Management (EMM) entity before the network has established a secure NAS signaling connection [3GP11c]:

- IDENTITY REQUEST (if requested identification parameter is IMSI)
- AUTHENTICATION REQUEST
- AUTHENTICATION REJECT
- ATTACH REJECT (if the EMM cause is not #25)
- DETACH ACCEPT (for non switch off)
- TRACKING AREA UPDATE REJECT (if the EMM cause is not #25)
- SERVICE REJECT (if the EMM cause is not #25)

Shaik et al. have suggested that the unprotected NAS signaling message listed above can be exploited in practical attacks such as location disclosure, Denial-of-Service (DoS) and forcing a victim to use the less secure GSM standard [SBA+15]. Chapter 3 explains how to exploit unprotected signaling messages to catch IMSIs and hence disclose the position of subscribers.

Paging is a signaling procedure integrated into any mobile systems. eNodeBs broadcast paging messages to all neighboring UEs unprotected. Chapter 4 describes how to exploit the unprotected paging messages to perform attacks against privacy in LTE. Accordingly, LTE has several known vulnerabilities, and practical attacks have successfully been implemented. Also, some of the security parameters are optional and often not enabled.

Chapter 3
Using IMSI Catchers

An IMSI Catcher is essentially a device that acts as a false base station used to collect IMSIs from surrounding UEs. This chapter explains how to use an IMSI Catcher for location disclosure and movement tracking. Additionally, this chapter describes the hardware and software tools used to build and configure an experimental IMSI Catcher, as well as a technical explanation of the attack and the results obtained. All the experiments described in this chapter have been performed multiple times to confirm the results.

IMSIs and other sensitive information gathered during the experiments are censored due to privacy concerns. As described in Section 2.6.2, IMSI is a unique private identification that should only be known by the associated PLMN.

3.1 Ethics / Privacy Concerns

The experiments in this chapter reveal vulnerabilities in the LTE network that can affect any LTE enabled device. UEs trying to connect to the IMSI Catcher is rejected with EMM rejection cause number 15 (*No suitable cells in TA*), to minimize service outage during the experiment [3GP11c]. Consequently, the UE will interpret the IMSI Catcher as unavailable and return to the commercial LTE network. Furthermore, sensitive information such as IMSI is censored to preserve subscriber identity confidentiality. The experiment was conducted late in the evening when there were few people close to the experiment, to prevent unauthorized UEs from connecting to the IMSI Catcher. Also, the output power of the USRP was reduced to 10dbm to limit the coverage area of the IMSI Catcher.

3.2 The Development of IMSI Catchers

An IMSI Catcher is a device perceived as a real BS by UEs, used to perform several attacks in GSM, UMTS, and LTE. Although the IMSI Catcher appears as a real BS,

it is not part of the infrastructure of a commercial PLMN. The main objective of the IMSI Catcher is to collect IMSIs from surrounding UEs. Since IMSI permanently identifies a UE, this sort of identity disclosure is critical with regard to user identity confidentiality. Consequently, by obtaining the IMSI of a UE, an adversary can track the movement of a subscriber over an extended period. Even more critical, IMSI Catcher applied in GSM networks allows an adversary to perform MITM attacks. By exploiting the vulnerability that GSM AKA does not support mutual authentication, an adversary can intercept all messages between the UE and the serving network.

Figure 3.1: Harris Corporation's first IMSI Catcher, the StingRay. Source: [Rya].

IMSI Catchers have existed for a long period, and the first IMSI Catcher was commercially launched by the German company Rohde & Schwarz in 1996 [Ooi15]. However, the most publicly known IMSI Catcher is the StingRay produced by Harris Corporation. The company made two models of the StingRay, the original StingRay sold for $68,479 USD and the StingRay II sold for $134,952 USD [Rya]. Figure 3.1 shows the first portable StingRay.

Today's IMSI Catchers are much handier in terms of price and size. The IMSI Catcher used to perform experiments in this thesis is run on the USRP B200mini produced by Ettus Research. The B200mini card is the size of a credit card, making it practical and discrete. Also, the prize has dropped drastically; the $134,952 USD StingRay II can be replaced by the B200mini costing $726 USD[1]. Another improved

[1]The price of the StingRay II also contains the software handling the IMSI catching logic, while the price of the B200mini only contains the hardware

feature of the B200mini is a convenient bus-powered Universal Serial Bus (USB) 3.0 connectivity. Section 3.3.3 describe the full specification of the B200mini.

3.3 IMSI Catcher Setup

3.3.1 Overview

This section describes how to set up an experimental IMSI Catcher, using the open source platform OpenAirInterface and the USRP B200mini. Notably, the B200mini handles the transceiving radio part, while OpenAirInterface handles all the logic. Below is a detailed list of all the necessary hardware and software that were used to build the IMSI Catcher and conduct the experiment:

- Desktop computer, Ubuntu 14.04 LTS, 3.19.0-031900-lowlatency kernel
  - Memory 8 GiB
  - Processor Inter Core i7 860 @ 2.80GHz * 8
  - Graphics GeForce GT 630
- OpenAirInterface running both openair-cn (core) and openairinterface5g (eNodeB)
- MySQL database v.5.5.52-0ubuntu0.14.04.1
- USRP B200mini-i, 70 MHz - 6 GHz frequency range, full duplex and USB 3.0 bus-powered
- Wireshark v1.10.6
- Atom - text editor v1.10.2

3.3.2 OpenAirInterface

OpenAirInterface is an open source platform developed by EURECOM, written in C [Ope16]. OpenAirInterface aims to follow the 3GPP standardization of the LTE protocol stack, to make the software structure similar to a commercial LTE network [Ope16]. The OpenAirInterface project is mainly divided into three projects: openairinterface5G (E-UTRAN), openair-cn (EPC), and oai1B (UE). Consequently, OpenAirInterface was chosen for this thesis because it supports multiple deployment scenarios and provides EPC, eNodeB and UE configurations. Deployment scenarios implemented in this thesis are the following:

- Commercial UE $\leftrightarrow$ OpenAirInterface eNodeB $\leftrightarrow$ OpenAirInterface EPC

– OpenAirInterface UE ↔ Commercial eNodeB ↔ Commercial EPC

The first deployment scenario is used in the IMSI Catching experiment, further explained in Section 3.5. The second deployment scenario is used to catch paging messages passively, further explained in Chapter 4. Furthermore, all OpenAirInterface entities can freely be cloned from `gitlab.eurecom.fr's` Github repository [Opea].

EURECOMs motivation behind creating OpenAirInterface was to simplify the current Radio Access Network (RAN) consisting of numerous elements that complicated and increased the cost for operators to deploy new services [Ope16]. Consequently, EURECOM created OpenAirInterface intended to simplify network access, facilitate application upgrades, and reduce cost [Ope16]. However, OpenAirInterface is still a work in progress and can not be compared to a full-fledged commercial Evolved Packet System (EPS) yet. The software lacks stability and tends to terminate sporadically without displaying any error message. One major drawback with OpenAirInterface is that it does not support voice communication nor Short Message Service (SMS). Although OpenAirInterface lacks some serious functionality, it manages to authenticate UEs and therefore considered feasible for this thesis.

3.3.3 USRP B200mini

B200mini is a USRP manufactured by Ettus Research, intended to be a low-priced device used by universities, research labs, and hobbyists [Nor06]. The B200mini utilizes a wide frequency range (70MHz - 6GHz), which allows operation of all frequency bands in GSM, UMTS, and LTE [Res]. Figure 3.2 illustrates the B200mini's small size, making it portable and physically hard to detect. Another great feature of the B200mini is the convenient bus-powered USB 3.0 connectivity, which allows it to be operated from anywhere with the help of a portable battery pack. The B200mini include several inputs and outputs, which is helpful during troubleshooting and connectivity check [Ett]:

– PWR LED (Power Indicator) The PWR LED is orange if the USRP is connected to power

– TRX LED (TX/RX Activity) The TRX LED is green if data is being received, red if data is being transmitted, and orange if the USRP alternating between transmitting and receiving

– RX2 LED (RX2 Activity) The RX2 LED is green if the USRP is receiving data

– SW1 (Hard Reset Switch) The SW1 is a switch used to reset the system

The B200mini is mounted with two antennas: one antenna used for both transmitting and receiving, while the other is used for receiving only. Both antennas are of type W1900, produced by Pulse Electronics [Pul].

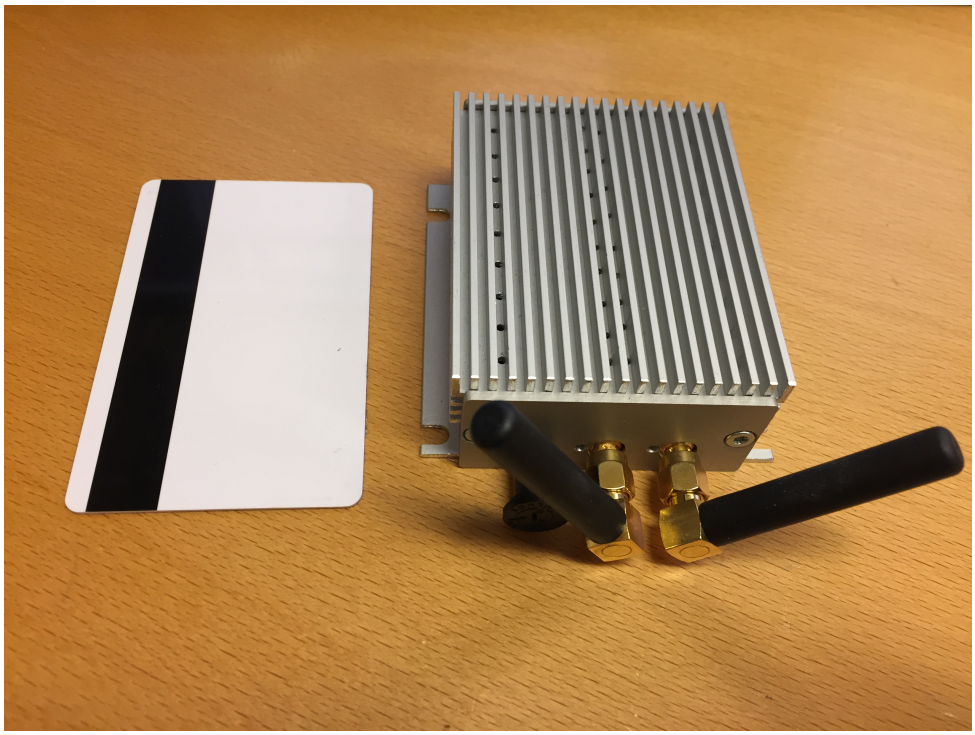

Figure 3.2: USRP B200mini with custom-made encapsulation. The B200min is placed next to a credit card to illustrate the small size.

3.3.4 Topology

USRP B200mini, host PC, Android mobile phone, and OpenAirInterface are the main components needed to set up an LTE IMSI Catcher. The OpenAirInterface software installed on the host PC constitutes the EPC, and the USRP connected to the eNodeB software constitutes the E-UTRAN. Figure 3.3 illustrates the full topology of the experimental setup. Notably, to provide IP connectivity to the UE, the host PC requires an Internet connection; however, during this experiment, the UE does not require IP connectivity.

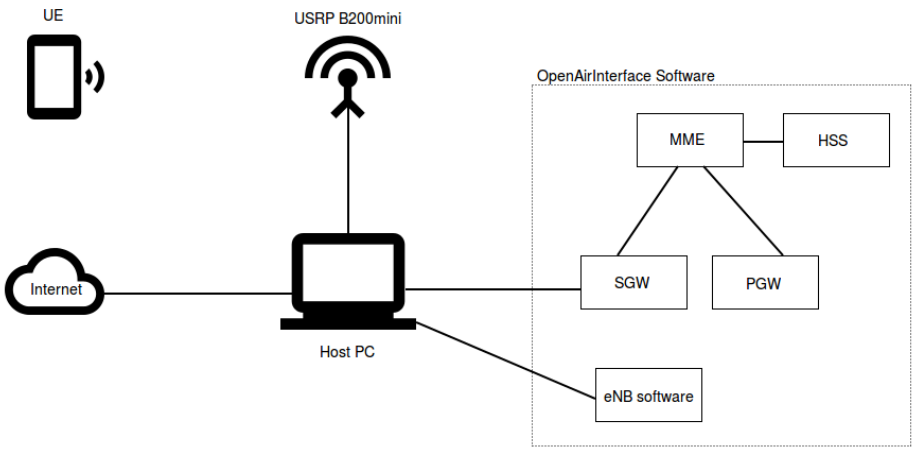

Figure 3.3: Topology of the LTE IMSI Catcher.

3.3.5 Wireshark

Wireshark is the software tool used to capture and intercept traffic between the UE, the MME, and the HSS. Since the UE, the MME, and the HSS are configured to communicate over the loopback interface, Wireshark can capture all package exchanges using the following command:

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

3.3.6 Set Up a Test Network Using OpenAirInterface

Before installing OpenAirInterface, is it important to make sure that the required hardware and the dependent software are in order. Section 3.3 provides a list of compatible hardware and required software. The initial step for setting up a test network is to clone and install the project from OpeAirInterface's web page [Ope17]. Subsequently, the MME, the HSS, and the Serving Gateway/PDN Gateway (SPGW) are built using the following commands in the terminal:

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

Subsequently, the eNodeB is built using the following commands in the terminal:

```
cd ~/openairinterface5g/cmake_targets
$ ./build_oai -w USRP -x -c --eNB
```

Note: By including the -i option OpenAirInterface will automatically install all missing packages. Also, note that the eNodeB entity should be built in the *openairinterface5g* directory.

The following commands are executed in the terminal to start each entity and deploy the IMSI Catcher:

```
$ ./run_hss
$ ./run_mme
$ ./run_spgw
$ sudo -E ./lte-softmodem -O enb.band7.tm1.usrpb210.conf -d
```

Consequently, all the running entities will together constitute an LTE test network (IMSI Catcher). Furthermore, the configuration parameters of the network can be changed to achieve the desired network. The default LTE test network is configured with MCC=208, MNC=93, and DL frequency=2680 MHz. Consequently, by choosing the country code of France, an unused network code, and a DL frequency that is not within the allocated frequencies of Norwegian PLMNs, interfering with commercial PLMNs are avoided. Appendix A gives a full description of prerequisites, configuration parameters, and commands needed to set up an LTE test network.

3.4 Catching IMSIs

As described in Section 3.2, IMSI Catchers have existed since GSM; however, IMSI catching in LTE is still an undiscovered research area. The following section describes how an LTE IMSI Catcher is used to collect surrounding IMSIs with low probability of detection by subscribers.

3.4.1 Overview

One Jammer and one Collector are needed to build an LTE IMSI Catcher operating in LTE networks. Essentially, both the Jammer and the Collector are two separated IMSI Catchers; however, they work together as one in the IMSI Catching process. Both the Jammer and the Collector are built using the topology depicted in Figure 3.3,

where B200mini and OpenAirInterface constitute the access network and the core network. The Jammer and the Collector are further described in Section 3.4.3.

3.4.2 Build an IMSI Catcher

The first functional requirement for an IMSI Catcher is to force UEs to attach to it. GSM connected UEs are regularly scanning for surrounding eNodeBs, if more than one eNodeB is detected, the one with the highest signal power is preferred. Strobel has proven that a GSM IMSI Catcher can exploit this feature [Str07]. By simply masquerade as a real BS and operate with the highest signal power, surrounding UEs automatically attaches to the IMSI Catcher [Str07]. However, this feature may not always be feasible in LTE. LTE enabled UEs located close to a serving BS already have sufficient signal power and excludes searching for surrounding BSs to save battery power. Necessarily, to overcome this obstacle the *absolute priority* feature is exploited. 3GPP LTE Release 8 specification first introduced the *absolute priority* feature [SBA+15, 3GP13].

Absolute Priority

UEs in *RRC IDLE* state periodically receives prioritized frequencies from the serving and neighboring eNodeBs. Moreover, the eNodeB performs reselection based on the *absolute priority*, which indicates that UEs always try to connect to the eNodeB with highest prioritized frequency [3GP10]. The PLMN is responsible for allocating the *absolute priorities*, which are only valid within the PLMN [3GP10]. The *absolute priorities* are attached to SIB type 4, 5, 6, and 7, and broadcasted by all eNodeBs [3GP16b]. Consequently, the highest prioritized frequency of a mobile operator can be obtained by passively sniffing SIB messages. Chapter 4 describes how to use a modified IMSI Catcher (SIB Catcher) to sniff SIB messages passively.

After obtaining the high priority frequencies, the next step is to masquerade as a commercial eNodeB. Consequently, by using the same MNC and MCC as a commercial PLMN, the IMSI Catcher impersonates the commercial eNodeB. Notably, a list of MCC and MNC values are publicly published yearly by the International Telecommunication Union (ITU) [Int16], alternatively; it can be retrieved by sniffing SIB type 1 messages from commercial eNodeBs. Table 3.1 provides a description of the SIB message types relevant to this thesis[2].

[2]Intra-frequency corresponds to frequencies in the same EUTRA band, and intra-frequency corresponds to frequencies in different EUTRA bands.

Table 3.1: System Information Block messages in LTE (excluding SIB 10-13) [3GP16b].

Message Type	**Description**
SIB type 1	Cell access information and scheduling of other SIBs
SIB type 2	Common and shared radio resource configuration for all UEs
SIB type 3	Cell re-selection parameters for intra-frequency, inter-frequency and/ or inter-RAT cell re-selection
SIB type 4	Information related to E-UTRAN intra-frequency neighboring cells
SIB type 5	Information related to E-UTRAN inter-frequency neighboring cells
SIB type 6	Information regarding inter-RAT cell re-selection (UTRAN cell information)
SIB type 7	Information regarding inter-RAT cell re-selection (GERAN cell information)
SIB type 8	Information regarding inter-RAT cell re-selection (CDMA2000 cell information)
SIB type 9	Information related to home eNodeB name

3.4.3 Jammer and Collector

The experiments in Chapter 4 have revealed that Norwegian PLMN tend to have eNodeBs operating with the highest prioritized frequency in urban areas, making it hard to exploit the *absolute priority* feature. However, two IMSI Catchers circumvent this problem. The two IMSI Catchers are from now on referred to as Collector and Jammer.

The Jammer has the responsibility to block the eNodeB operating with the highest prioritized frequency. The Jammer simply operates on the same frequency as the commercial eNodeB, causing interference between the two eNodeBs. Meanwhile, the Collector operates with the second highest prioritized frequency and have the responsibility to perform the actual IMSI acquisition. Since the Jammer blocks the highest prioritized frequency, UEs will automatically try to connect to the second highest prioritized frequency, which is the Collector. It is important that the Collector is the first one turned on, followed by the Jammer. Otherwise, the UE may bypass the Collector and directly connect to another commercial cell.

Moreover, if a UE detects a new TA, it sends a *"TAU request"* to the eNodeB. Consequently, to trigger the *TAU request* message from the UE, the IMSI Catcher must be configured with a TAC that is different from the commercial eNodeB [SBA+15]. The TAC of the commercial eNodeB can be retrieved by sniffing SIB type 1 message. Section 3.5.3 discuss the TAU procedure further.

3.4.4 Jammer and Collector Configurations

As described in Section 3.4.3, the Jammer is used to interfere with the prioritized frequency while the Collector is used to acquire IMSIs. There are two ways to find the configuration parameters for the Jammer and the Collector: by using pre-installed smartphone software or by using a USRP. Table 3.2 summarizes the configuration parameters employed in the IMSI Catcher experiment.

Using iPhone to Determine Configuration Parameters

The following method only requires an iPhone and a valid LTE subscription[3]; however, the method is only able to obtain configuration parameters for one PLMN. Most iPhone models contain a graphical user interface to view the "Field Test Mode", which is a menu providing technical information about serving and surrounding cells. The "Field Test Mode" is invoked by dialing *"*3001#12345#**"*. The highest prioritized frequency is most likely the frequency of the serving cell and can be found by navigating to *"Serving Cell Measurements"* in the *"Field Test"* menu. Figure 3.4 depicts the complete *"Field Test"* menu.

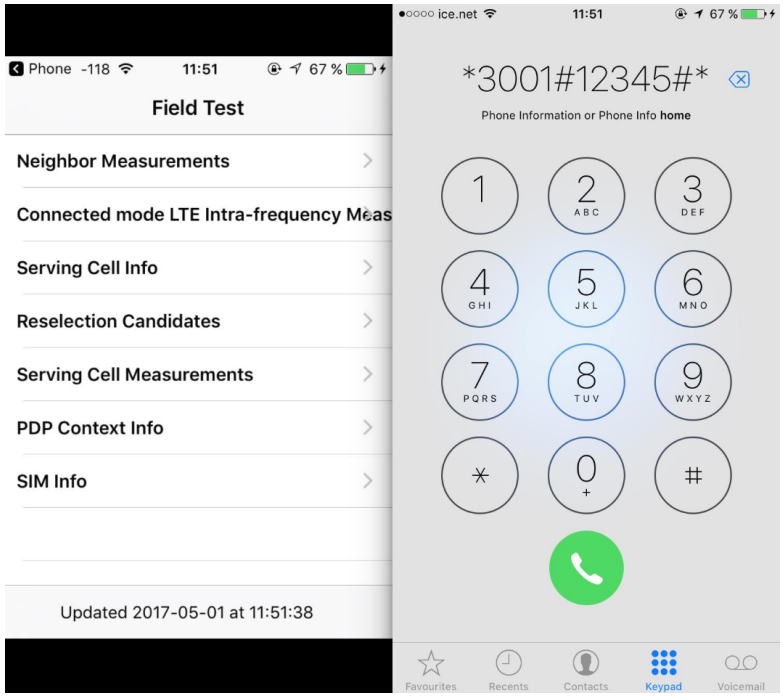

Figure 3.4: *Field Test* menu in iPhone.

[3]The method is also applicable to other smartphone models [MO17].

The *"Serving Cell Measurements"* menu only provides the E-UTRA Absolute Radio Frequency Channel Number (EARFCN) of the serving cell; consequently, equation 1 and 2 calculates the DL and the UL frequency. The DL frequency is calculated using equation (3.1), where N_{DL} is the DL EARFCN, and the 3GPP standard defines F_{DL_low} and $N_{offs-DL}$ [3GP11a].

$$F_{DL} = F_{DL\_low} + 0.1(N_{DL} - N_{offs-DL}) \tag{3.1}$$

The UL frequency is calculated using equation (3.2), where N_{UL} is the UL EARFCN, and the 3GPP standard defines F_{DL_low} and $N_{offs-DL}$ [3GP11a].

$$F_{UL} = F_{UL\_low} + 0.1(N_{UL} - N_{offs-UL}) \tag{3.2}$$

Further, the configuration parameters for the Collector are obtained by navigating to *"Neighbor Measurements → Neighbor Cells List"* in the *"Field Test"* menu. The list does not contain the priority of the frequencies; consequently, the trial and fail method determine the frequency for the Collector.

Using USRP to Determine Configuration Parameters

The following method is more comprehensive and requires additional software and hardware tools; however, it obtains configuration data for multiple PLMNs. Necessary hardware and software are USRP, PC with Ubuntu 14.04, and OpenAirInterface. Notably, OpenAirInterface is configured as a UE, used to catch and decode SIB type 5 messages. As described in Table 3.1, the SIB type 5 message contains the priority of neighboring cells operating in different frequency bands. Consequently, by comparing the priority for each cell, the configuration parameter for the Jammer and Collector are determined[4]. Section 4.4.5 describes how to configured OpenAirInterface as a UE and how it can be used to catch SIB messages from commercial eNodeBs.

Both of the methods confirmed the configuration parameters for the Jammer and the Collector. Table 3.2 summarizes the configuration parameters used in the experiment conducted in Section 3.5. Accordingly, the Jammer is configured to operate in band 3 with EARFCN 1650, while the Collector is operating in band 20 with EARFCN 6300.

[4]'finnsenderen.no' is a tool that shows the geographical location for all the eNodeBs in Norway, and might help to give a better understanding [Nko].

Table 3.2: Configuration parameters for the Collector and the Jammer.

IMSI Catcher	Band	DL Frequency (MHz)	UL Frequency (MHz)	EARFCN
Collector	20	806	847	6300
Jammer	3	1850	1755	1650

3.5 Experiment

The following IMSI Catcher experiment was conducted in the Electrical engineering building at Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. The objective of the experiment was to use an IMSI Catcher to spoof Telia and collect IMSIs[5]. Furthermore, Appendix B contains the code changes required to achieve the IMSI Catcher behavior.

3.5.1 Overview

Ideally, the IMSI Catcher should be able to catch IMSIs without the subscriber noticing it. Consequently, after receiving the IMSI, the Collector indicates to the UE that it is no longer available, causing the UE to reconnect to a commercial eNodeB. The experiment consists of three phases:

- The Jammer blocks the prioritized frequency of the commercial LTE network, causing UEs to disconnect from the serving cell and start searching for new cells
- The Collector attracts searching UEs and steal their IMSI
- The UEs eventually disconnects from the Collector and returns to the commercial network

Figure 3.5 summarizes the flow of the experiment. Moreover, the second phase of the experiment represents message 1-7 in Figure 3.5, while the third phase of the experiment represents the eighth message[6].

3.5.2 Configurations

As mentioned in section Section 3.4.2, to be able to spoof a commercial PLMN, the same MCC and MNC are used. Consequently, the following configurations were applied to the Collector:

[5]Telia was chosen for this experiment because the available test UEs had Telia subscriptions.

[6]The first phase of the experiment did not contain any message exchanges and hence excluded from Figure 3.5.

- Select the MCC of Norway: MCC=242
- Select the MNC of Telia: MNC=02
- Select a TAC different from the Telia TAC in the location of the experiment: TAC=01
- Choose the second highest prioritized frequency: 806MHz

Moreover, the Jammer has the same configuration as the Collector, except the frequency, which is 1850MHz. Table 3.2 summarizes the configuration parameters for the Jammer and the Collector.

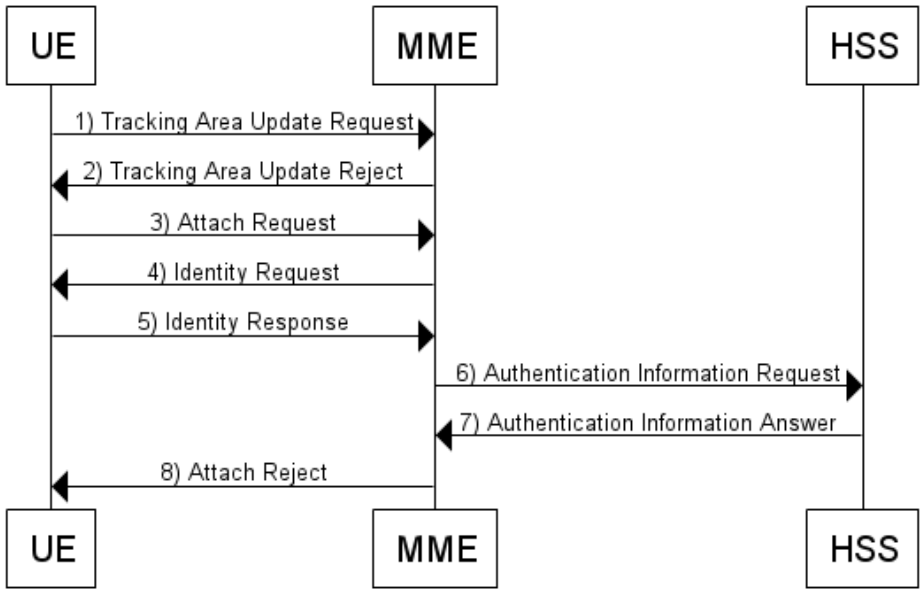

Figure 3.5: LTE IMSI Catcher (Collector) message exchange.

3.5.3 TAU Procedure

The TAU procedure is always initiated by the UE and may be triggered by several events. The Collector is configured with a different TAC than the commercial eNodeB. As a result, the UE triggers a *TAU Request* due to the TAC of the Collector is not in the list of previously registered TACs [3GP11c]. Subsequently, a *TAU Reject* message is sent to the UE, dismissing the TAU procedure. As illustrated in Figure 3.6 the *TAU Reject* message contain EMM rejection cause 10 (*implicitly detached*), indicating that the UE is entering the *deregistered* state and immediately performs a

new attach procedure [3GP14c]. Message 1 and 2 in Figure 3.5 illustrates the *TAU Request* and *TAU Reject* message. Moreover, the TAU procedure does not contain any relevant information for this experiment; however, it can be used to check for GUTI persistence [7].

```
▾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
      ▸ProtocolIE-Field
     ▾Item 2: id-NAS-PDU
      ▾ProtocolIE-Field
        id: id-NAS-PDU (26)
        criticality: reject (0)
       ▾value
        NAS-PDU: 074b0a
       ▾Non-Access-Stratum (NAS)PDU
        0000 .... = Security header type: Plain NAS message, not security protected (0)
        .... 0111 = Protocol discriminator: EPS mobility management messages (0x07)
        NAS EPS Mobility Management Message Type: Tracking area update reject (0x4b)
       ▾EMM cause
        Cause: Implicitly detached (10)
```

Figure 3.6: Wireshark capture of a *TAU Reject* message returning EMM rejection cause 10.

3.5.4 Attach Procedure

The attach procedure is the most valuable part of this experiment, represented as message 3-8 in Figure 3.5. UEs initiates the attach procedure to connect to a network and use the services provided by the EPC [3GP11c].

An *Attach Request* message is sent by the UE to the MME to initiate the attach procedure. The *Attach Request* contain valuable information such as the *EPS mobile identity* containing the IMSI, the GUTI, or the IMEI [3GP11c]. For the sake of subscriber identity confidentiality the *EPS mobile identity* ideally contain the temporary identity, the GUTI. However, the Collector can force the UE to reveal its permanent identity (IMSI) by initiate an *Identity Request* message. The *Identity Request* is used to inform the UE that the serving network (Collector) is unable to retrieve the IMSI from the GUTI, and hence requests the IMSI. Figure 3.7 highlights how the IMSI Catcher uses the *Identity Request* to obtain the IMSI.

[7] Shaik et al. have discovered that network operators tend not always to update the GUTI, and UEs may have the same GUTI for up to three days [SBA+15].

```
▾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 (0x07)
     NAS EPS Mobility Management Message Type: Identity request (0x55)
     0000 .... = Spare half octet: 0
     .... 0001 = Identity type 2: IMSI (1)
```

Figure 3.7: *Identity Request* message initiated by the IMSI Catcher to obtain the IMSI.

Upon reception of the *Identity Request*, the UE interprets the message and returns the requested IMSI in an *Identity Response* message. Figure 3.8 depicts how a UE provides its IMSI to the Collector unprotected. The IMSI is censored to preserve subscriber identity confidentiality; however, the first five uncensored digits of the IMSI reveals that the subscriber is located in Norway and have a Telia subscription. As listed in Section 2.7, the *Identity Request* message does not require a secure exchange between the UE and the network. Consequently, the Collector send and unprotected *Identity Request* to the UE, resulting in an unprotected *Identity Response* from the UE.

```
 NAS-PDU: 1736270988090756082924200980571589
▾Non-Access-Stratum (NAS)PDU
  0001 .... = Security header type: Integrity protected (1)
  .... 0111 = Protocol discriminator: EPS mobility management messages (0x07)
  Message authentication code: 0x36270988
  Sequence number: 9
  0000 .... = Security header type: Plain NAS message, not security protected (0)
  .... 0111 = Protocol discriminator: EPS mobility management messages (0x07)
  NAS EPS Mobility Management Message Type: Identity response (0x56)
 ▾Mobile identity - IMSI (24202[redacted])
   Length: 8
   0010 .... = Identity Digit 1: 2
   .... 1... = Odd/even indication: Odd number of identity digits
   .... .001 = Mobile Identity Type: IMSI (1)
   BCD Digits: 24202[redacted]
```

Figure 3.8: *Identity Response* message containing the IMSI.

Message 6 and 7 in Figure 3.5 illustrates the *Authentication Information Request* and *Authentication Information Answer* message. Upon reception of the IMSI, the

MME encapsulates the IMSI in an *Authentication Information Request* message and forwards it to the HSS. Consequently, the HSS have the responsibility to check if subscription data exists for the IMSI [3GP11b]. Notably, the Collector does not need to authenticate the UE in this experiment and responds that it does not have a pre-computed Authentication Vector (AV) available [3GP11b]. The response is sent as an *Authentication Information Answer* and contains the result code *DIAMETER_AUTHENTICATION_DATA_UNAVAILABLE*.

The final step of the attach procedure is to disconnect the UE. Since the Collector already has obtained the IMSI, it sends an *Attach Reject* message to the UE. Consequently, the *Attach Reject* message indicates that the network has rejected the *Attach Request* [3GP14c]. The *Attach Reject* message may contain different EMM rejection causes, used to inform the UE why the network rejected it. Appendix C provides a complete list of all the possible rejection values that can occur in an *Attach Reject* message. Moreover, the Collector is configured to return EMM rejection cause #15 *"No suitable cells in TA"*, illustrated in Figure 3.9. Rejection cause #15 instructs the UE to store the TAI of the Collector in the list of *"forbidden tracking areas for roaming"* and enter the *EMM-DEREGISTERED* state [3GP11c]. Subsequently, the UE start searching for other suitable cells in another tracking area in the same PLMN [3GP11c]. Since the TAI of the Collector exists in the *"forbidden tracking areas for roaming"* list, the UE will not attempt to attach to the Collector until the UE restarts or loses power.

```
▾Item 2: id-NAS-PDU
 ▾ProtocolIE-Field
   id: id-NAS-PDU (26)
   criticality: reject (0)
  ▾value
    NAS-PDU: 07440f
   ▾Non-Access-Stratum (NAS)PDU
     0000 .... = Security header type: Plain NAS message, not security protected (0)
     .... 0111 = Protocol discriminator: EPS mobility management messages (0x07)
     NAS EPS Mobility Management Message Type: Attach reject (0x44)
    ▾EMM cause
     Cause: No Suitable Cells In tracking area (15)
```

Figure 3.9: *Attach Reject* message returning EMM rejection cause #15 (No Suitable Cells In Tracking Area).

Consequently, by applying EMM rejection cause #15 to the *Attach Reject* message, the Collector transfers the UE back to the commercial network with a low probability of the subscriber noticing the IMSI catching. The experiment has shown that the average time between the *Attach Request* message and the *Attach Reject* message is 23.5622 milliseconds; i.e. the UE is disconnected from the commercial network and

loses LTE service for 23.5622 milliseconds in addition to the time it takes to reattach to the commercial network. Due to the automatic reattachment and the short service disruption, subscribers with a regular LTE subscription will have difficulties noticing the IMSI catching, unless he/she is in a phone call.

Appendix D contains calculations of the average time between an *Attach Request* and an *Attach Reject* message for the Collector.

Results

The Collector was spoofing Telia for 35 seconds and was able to catch 3 different IMSIs. Retterstøl conducted a similar experiment in 2015 where he spoofed NetCom's (former Telia) GSM network [Ret15]. Retterstøl managed to capture 8 IMSIs in 11 minutes. As a result, the IMSI Catcher setup explained in thesis can be considered far more efficient than the IMSI Catcher introduced by Retterstøl. Since both the experiment in this thesis and Retterstøl's experiment were conducted on the same network operator and in the same location, the experiments are considered comparable[8]. Table 3.3 summarizes the results obtained by the Collector. Notably, one of the captured IMSIs had the MNC used for Telia's subsidiaries (05), while two IMSI had Telia's original MNC (02). Also, all the collected UEs were previously attached to the same commercial cell; this behavior was expected since the Jammer blocked the frequency of that cell. The S-TMSIs were retrieved from the *TAU Request* messages.

Table 3.3: IMSIs obtained when spoofing Telia. MSINs are censored.

IMSI	**Previous Cell Identity**	**S-TMSI**
24202XXXXXXXXXX	15597824	0xc4004890c4
24202XXXXXXXXXX	15597824	0xe4007a0324
24205XXXXXXXXXX	15597824	0xc4005713c2

3.6 Use IMSI for Location Disclosure

The experiment in Section 3.5 revealed that catching IMSIs in LTE networks is feasible. The IMSI is stored on the UICC and cannot be changed unless the UICC card is physically replaced. IMSI disclosure is a severe violation of subscriber identity confidentiality and may have fatal consequences. As a result, the IMSI can be used to reveal subscriber location and track movement over time.

[8] A conclusion would require much more data from both experiments.

3.6.1 UE Positioned in Cell Coverage Area

Having retrieved the IMSIs using the Collector, it is necessary to associate each IMSI to a subscriber. Subsequently, UE location disclosure can be achieved using three different techniques [SBA+15]. The first technique is utilized by the Collector to check whether a UE's IMSI is in the list of collected IMSIs. If the IMSI is in the list of collected IMSIs is it safe to claim that the UE is located within the coverage area of the Collector. Factors such as geographical conditions and propagation conditions influence the coverage area [Sha16]; however, the coverage area of an LTE cell in urban areas is typically 2 km² [SBA+15]. The second technique determines the UE's previous and future occurrences in a particular area [SBA+15]. The last technique is used to ensure that a UE is not located within a certain area. Consequently, the Collector is searching for a specific IMSI within its coverage area. This technique cannot be used to locate a UE accurately; however, it can be used to perform malicious activities, for example, a burglar can use this information to determine if a particular subscriber is located inside a particular building.

3.6.2 UE Positioned in Expanded Cell Coverage Area

Shaik et al. have determined the average cell radius in cities to be 800 meters for the 2.6 GHz frequency band and 1 km for the 800 MHz frequency band [SBA+15]. Moreover, the B200mini used in this experiment is transmitting with a maximum output power of 100mW [Mat], which equals a cell radius up to 100 meters [SBA+15]. However, a power amplifier can extend the coverage sufficiently[9]. As the transmission distance increases, the three location disclosure techniques described in Section 3.6.1 will become very inaccurate.

The previous connected commercial cell is used to improve the subscriber location accuracy for a Collector operating with a large coverage area. By using 'finnsenderen.no' and one of the two methods explained in Section 3.4.4, the geographical location and the coverage area of the commercial cell is determined. Consequently, the location of the subscriber lies within the intersections of the Collector and the commercial cell's coverage area, as illustrated in Figure 3.10. The yellow part of the figure highlights the coverage area of the Collector, the red part highlights the coverage area of the commercial cell, and the part where the circles overlap highlight the subscriber location[10]. Notably, the accuracy of this method decreases as the distance between the Collector and the commercial cell decreases. Also, if the distance between the Collector and the UE becomes too big, the UE might reselect another cell due to stronger received signal power.

[9]The COST-231 model is used to calculate the transmission distance for LTE band 3, and the HATA model is used to calculate the transmission distance for LTE band 20 [Sin12].

[10]The software tool SPLAT! can be used to calculate the coverage area of a cell based on the terrestrial conditions of a given area [SPL].

The test scenario illustrated in Figure 3.10 shows that an IMSI Catcher using a power amplifier can achieve a cell radius of 800 meters. The commercial cell is operating in the 800MHz frequency band and hence have a cell radius of 1 km. Consequently, an IMSI Catcher located at "Marinen" (center of yellow circle) is able to catch IMSIs located at Gløshaugen campus. The setup illustrated in Figure 3.10 have not been tested in this experiment; however, Section 3.4.2 suggest that IMSI Catchers with extended coverage should be theoretically feasible to implement.

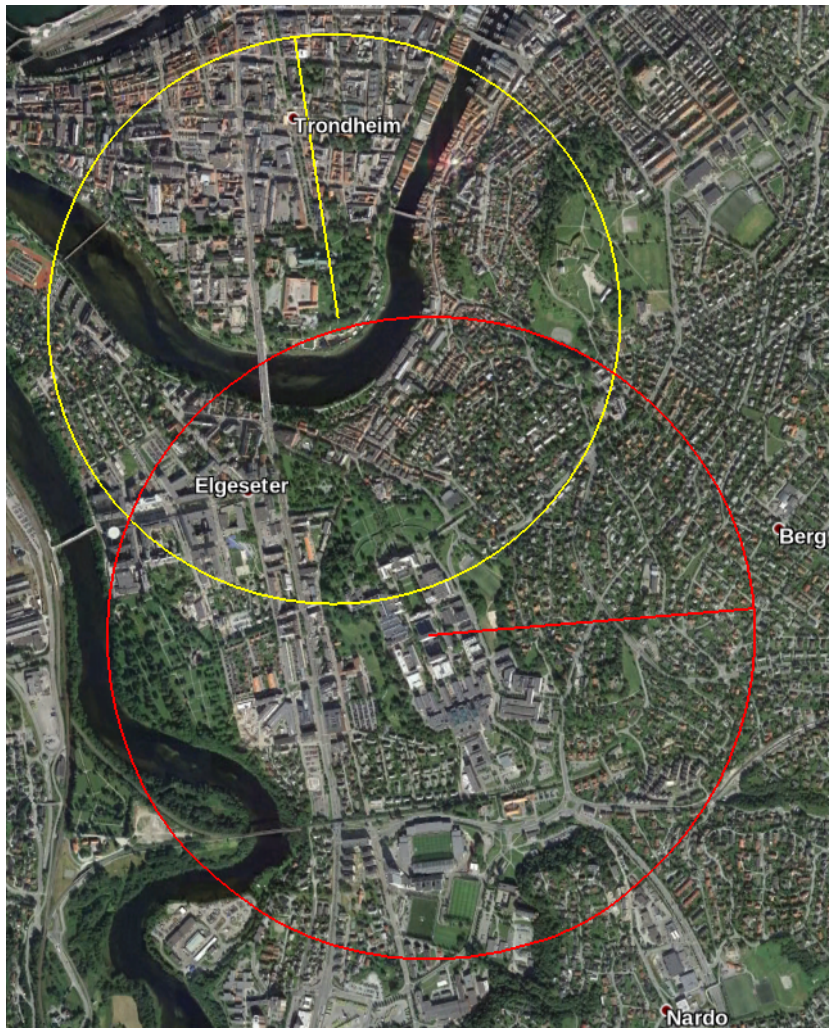

Figure 3.10: Map of the coverage area of the Collector and the commercial cell. The yellow circle highlights the coverage area of the Collector and the red circle highlights the coverage area of the commercial cell. Edited map from Google Earth Pro [Goo].

3.7 Countermeasures

Ooi has proposed several countermeasures against IMSI Catchers in GSM networks which are also applicable for LTE IMSI Catchers [Ooi15].

3.7.1 Unregistered Cell ID

As mentioned in Section 3.5.3 to trigger a *TAU Request* message from the UE, the IMSI Catcher must be configured with a different TAC than the commercial eNodeB. As a result, by cross-checking the TAC issued by the network with a database containing legitimate base stations, the UE would know if the BS is real. Such a database is provided by Federal Communications Commission [Fed].

3.7.2 IMSI Catcher Catcher

By using tools which can catch IMSI Catchers, no changes in the LTE protocol are required. SnoopSnitch is a free Android app used to detect IMSI Catchers [Sno].

3.8 Discussion and Results

The experiment described in Section 3.5 was conducted in the wireless security lab at NTNU, and have proved that IMSI catching in LTE networks is feasible.

The experiment was conducted late in the evening when there were few people at the university to prevent unauthorized UEs from attaching to the IMSI Catcher. Also, the output power of the USRP was reduced to 10dbm to limit the coverage area of the IMSI Catcher. However, one of the captured IMSIs was not part of the experiment but was accidentally located close to the IMSI Catcher. As described in Appendix D the unauthorized UE was only attached to the IMSI Catcher for 0.017320 milliseconds, causing a minimal service disruption.

There were only used two test UEs in the experiment; however, the experiment was repeated several times to make sure the results were consistent. Consequently, by only having two test UEs, the IMSIs were quickly obtained, and thus the experiment was rapidly conducted each time. Furthermore, the two test phones were of the type LG Nexus 5X with a prepaid Telia subscription.

From Figure 3.4 it can be observed that some phone manufacturers reveals very detailed information about the serving cell and neighboring cells. This information can be useful for determining the configuration parameters for the Jammer and Collector in a given area.

Based on the results gathered in Section 3.5.4, the IMSI Catcher was able to catch three different IMSIs within 35 seconds. Even though there is insufficient data to draw a conclusion, the experiment indicates that the IMSI catching process is very efficient. Retterstøl conducted a similar IMSI Catcher project in 2015 where IMSIs in the GSM network was collected [Ret15]. Retterstøl was able to catch eight different IMSIs within eleven minutes, which indicates that the IMSI catcher introduced in this experiment appears to be more effective. Also, Retterstøl's experiment consisted of 31 participants, while this experiment was conducted late in the evening with only two participants.

Generally, UEs connected to a mobile network will prefer attaching to the BS providing the strongest signal power. However, this is not always the case in LTE. If a UE already is close to a serving BS and receives sufficient signal power, it will skip searching for other BSs. Consequently, to bypass this problem, Section 3.4.2 introduces the priority based cell reselection feature, which implies that a UE should periodically search for eNodeBs operating with a higher prioritized frequency. Remarkably, the *absolute priority* might be the reason why the LTE IMSI Catcher explained in this thesis appears more efficient that previous IMSI Catchers. Since the *absolute priority* is attached to SIB message 4-7, one can calculate how frequent the UE receives the *absolute priority*. As illustrated in Figure 3.11, the periodicity of SIB type 3-7 can be obtained by catching and decoding a SIB type 1 message (Chapter 4 explains how to catch and decode SIB messages). One can observe from the figure that the window length is 10 milliseconds and the periodicity for SIB type 4-7 is 64 radio frames. Since the window length is 10 second and contains one radio frame, UEs will receive SIB 4-7 once every 640 milliseconds from the cell analyzed in Figure 3.11.

```
904 <schedulingInfoList>
905     <SchedulingInfo>
906         <si-Periodicity><rf8/></si-Periodicity>
907         <sib-MappingInfo><sibType3/>
908         </sib-MappingInfo>
909     </SchedulingInfo>
910     <SchedulingInfo>
911         <si-Periodicity><rf64/></si-Periodicity>
912         <sib-MappingInfo><sibType4/>
913         </sib-MappingInfo>
914     </SchedulingInfo>
915     <SchedulingInfo>
916         <si-Periodicity><rf64/></si-Periodicity>
917         <sib-MappingInfo><sibType5/>
918         </sib-MappingInfo>
919     </SchedulingInfo>
920     <SchedulingInfo>
921         <si-Periodicity><rf64/></si-Periodicity>
922         <sib-MappingInfo><sibType6/>
923         </sib-MappingInfo>
924     </SchedulingInfo>
925     <SchedulingInfo>
926         <si-Periodicity><rf64/></si-Periodicity>
927         <sib-MappingInfo><sibType7/>
928         </sib-MappingInfo>
929     </SchedulingInfo>
930 </schedulingInfoList>
931 <si-WindowLength><ms10/></si-WindowLength>
```

Figure 3.11: SIB type 1 message containing periodicity for SIB type 3-7.

Chapter 4
Passive Broadcast Catcher

This chapter explains how to sniff and decode broadcast messages sent by commercial eNodeBs passively. SIB and paging messages are technically described and analyzed for each PLMN. Additionally, the chapter contains a description of the software and hardware tools used in the experiment. Furthermore, an in-depth analysis of ice.net's temporary identity allocation.

Chapter 2 provides relevant information for this chapter. Section 2.4 include descriptions of channels types used to transport and distribute broadcast messages to UEs, while Section 2.6 contains a description of the LTE security architecture and security aspects. Other relevant references will be noted during the chapter.

4.1 Ethics / Privacy Concerns

The experiments conducted in this chapter reveals vulnerabilities in LTE network that can affect any LTE enabled UE. Experiments performed in real LTE networks were carefully tested to avoid service disruption. Moreover, the experiments passively monitor network traffic without interrupting normal service. Also, the coverage area of the USRP was reduced to limit the amount of affected eNodeBs. Temporary identities are kept uncensored due to the expired validity period.

4.2 Paging

4.2.1 Paging Procedure

Paging is a procedure used by LTE networks to notify one or more UEs [3GP14b]. If the core network receives an IP packet from a network unknown to the P-GW, it broadcasts a paging message to all surrounding UEs, notifying them about the incoming IP packet [Rao11]. The addressed UE is the only one responding to the paging message. Moreover, the temporary identity S-TMSI or the permanent identity

IMSI identifies the paging message [3GP16b]. Figure 4.1 shows the paging procedure and the subsequent RRC connection establishment.

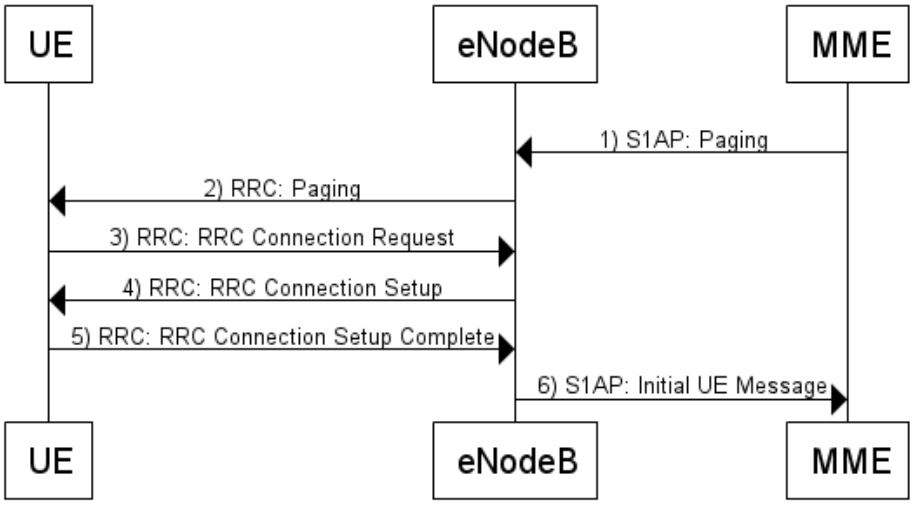

Figure 4.1: Paging procedure and successful RRC connection establishment.

When the MME receives an IP packet from an external network, it has to ensure that the UE have established a valid RRC connection. Consequently, the EPC uses smart paging to locate the subscriber[1]. Moreover, the MME sends an S1AP paging message to the eNodeB where the UE was last attached. If the UE does not reply, the paging message is broadcasted by every eNodeB in the TA [Dav13, SBA+15].

Message 1 in Figure 4.1 illustrates the S1AP paging message. Subsequently, the eNodeBs broadcast an RRC paging message to all the surrounding UEs, notifying them about the incoming IP packet [Rao11]. Each UE interprets the packets and discards it unless the *ue-Identity* matches the identity of the UE. Once the intended UE receives the RRC paging message from the eNodeB, it initiates the RRC establishment procedure, illustrated as message 3-5 in Figure 4.1 [Rao11]. The procedure is used to establish a valid RRC connection and let the UE inform the network which services it requests [3GP16b]. Message 3 indicates the *RRC Connection Request*, containing a reason why the UE wants to connect to the network. *RRC Connection Setup*

[1]Smart paging is used to reduce the signaling load in LTE networks by limiting the paging to one eNodeB [Dav13].

provides configuration parameters for a Signalling Radio Bearer (SRB), used to transport RRC messages between the UE and the eNodeB [3GP16b]. Eventually, the UE indicates to the eNodeB that the RRC connection establishment procedure has been completed and sends the initial UE message.

4.2.2 Paging Message Types

The following events may trigger the paging procedure [3GP16b]:

- *PagingRecord* - Transmit paging information to a UE, commonly triggered by voice calls or text messages. Notably, this information is highly relevant for the experiments conducted in Section 4.5 and Section 4.6
- *systemInfoModification* - Inform UEs about system information changes. Moreover, the paging messages do not provide the actual changes; it instructs the UE to re-acquire system information
- *etws-Indication* - Inform UEs about an Earthquake and Tsunami Warning System (ETWS) notification
- *redistributionIndication* - Instruct UEs to perform E-UTRAN inter-frequency measurement [3GP16c]

In most cases, paging messages are triggered by voice calls, SMS messages or other similar procedures. In urban areas, the paging occasion may be frequent, and an eNodeB may send multiple messages per second. Consequently, to reduce the number of messages, the eNodeB may address multiple UEs to the same paging message [3GP16b]. Implying that a paging message contains multiple *PagingRecords*, where each *PagingRecord* includes a UE identity. The paging message may also include non-subscriber related information, such as system information updates and/or ETWS notification. Table 4.1 summarizes the RRC paging message structure.

Table 4.1: RRC paging message structure [3GP16b].

Paging Field Descriptions	**Value**	
Paging Record List	Paging Record (1 to 16)	UE Identity (S-TMSI or IMSI)
System Information Modification	TRUE or FALSE	
ETWS Indication	TRUE or FALSE	

4.2.3 UE Identity

The 3GPP standardization defines that a *pagingRecordList* may contain up to sixteen instances of *pagingRecord*, whereas each *pagingRecord* contains one instance of *ue-Identity* [3GP16b]. Furthermore, *ue-Identity* uses the permanent identity IMSI or the temporary identity S-TMSI to identify a particular UE. Whether the IMSI or the S-TMSI is used depends on the PLMN. Ideally, an LTE PLMN preferably chooses the S-TMSI to preserve the subscriber identity confidentiality [Shab]. However, if the network fails to allocate a valid S-TMSI for the UE, the paging message is sent using the IMSI. Furthermore, if a UE receives a paging message containing IMSI, existing SRBs should be torn down, delete all associated security keys and perform a new attach procedure [3GP11c, Shab].

4.3 System Information

4.3.1 Overview

eNodeBs frequently broadcasts system information to all surrounding cells. The system information provides valuable information to the UE, such as cell re-selection, cell accessibility, intra-frequency, and inter-frequency. Furthermore, the system information is divided into MIB and multiple SIBs, all transported over the logical channel BCCH [3GP16b]. Table 3.1 provides a summary of the most relevant SIB messages for this thesis.

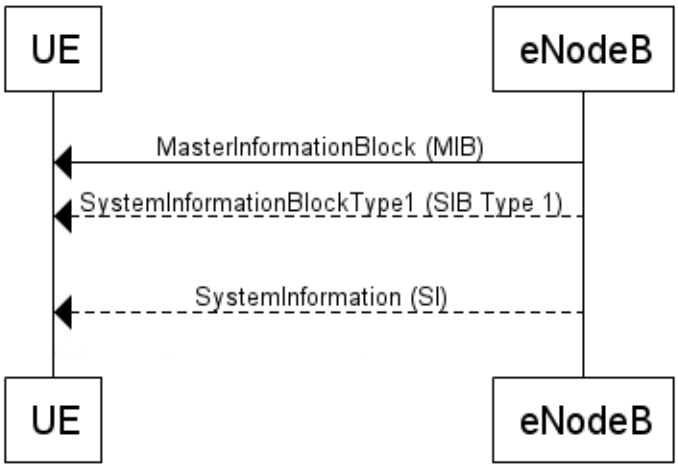

Figure 4.2: System information acquisition.

4.3.2 Master Information Block (MIB)

Figure 4.2 shows that the MIB is the first system information message interpreted by the UE, containing essential parameters required to obtain further information from the cell. The MIB content is static and broadcasted to all surrounding UEs once every 40 milliseconds[2] [3GP16b]. The BCH channel transmits the MIB and contains three fields [3GP16b]:

- *dl-Bandwidth* - Provides the transmission bandwidth configuration of the cell. This information is crucial, as it is used to decode other physical layer channels
- *phich-Config* - Provides the UE with information on how to decode the Physical Downlink Control Channel (PDCCH) channel [Nit16]
- *systemFrameNumber* - Defines the System Frame Number (SFN), used to obtain initial and periodic synchronization between the UE and the eNodeB [Nit16]

4.3.3 System Information Block (SIB)

SIB consists of thirteen different types, carried in System Information (SI) messages. SIB type 1 is not part of the SI messages and sent independently of the other SIBs. SI messages carry one or more SIBs, used to provide relevant information for the UE; however, SI message only carries SIBs with the same periodicity [3GP16b]. Moreover, each SIB contains a specific bulk of information, transmitted to a set of UEs. The transport channel DL-SCH transmits all SIB types [3GP16b].

SIB Type 1

SIB type 1 is one of the most important messages, as all the other SIB types depend on the information carried in this message. SIB type 1 is broadcasted once every 80 millisecond and stores the *schedulingInfoList*, carrying information about the periodicity of other SIs [3GP16b]. Additionally, SIB type 1 contains cell id, MNC, MCC, TAC, a mapping of SIBs to SI messages, and cell access restrictions [3GP16b].

Moreover, SIB 1 and SIB 2 plays a central role in the attach procedure. Before a UE can initiate an attach procedure, the information provided in SIB type 1 and SIB type 2, must be received and interpreted [Ari13].

SIB Type 2

The *schedulingInfoList* does not contain the periodicity of SIB type 2; however, the periodicity of SIB 2 always corresponds to the first entry in the *schedulingInfoList*

[2]The first MIB is transmitted in subframe #0 for which the SFN mod 4 = 0 [3GP16b].

[3GP16b]. The SIB type 2 contains RRC information common for all UEs, in addition to timers, UL power controls and configuration information for the logical channels PCCH and BCCH.

SIB Type 3-7

The *schedulingInfoList* in SIB 1 contains the scheduling periodicity for SIB 3-7. SIB 3-7 are all RRC information elements providing information regarding cell re-selection, neighboring cells, and intra/inter-frequencies [3GP16b].

4.3.4 Radio Network Temporary Identifier

Radio Network Temporary Identifier (RNTI) identifies the type of information intended for a particular UE. There exists several RNTIs, whereas only the Paging RNTI (P-RNTI) and the System Information RNTI (SI-RNTI) will be discussed in this thesis. The SI-RNTI addresses SIB type 1 and all SI messages, while P-RNTI addresses paging and system information changes [3GP16b, 3GP16a]. Both the SI-RNTI and the P-RNTI identifies the type of broadcast information. Table 4.2 summarizes the two identifiers.

Table 4.2: P-RNTI and SI-RNTI usage [3GP16a].

RNTI	Description	Transport Channel	Logical Channel	Value (hex)
P-RNTI	Paging and System Information change	PCH	PCCH	FFFE
SI-RNTI	Broadcast of System Information	DL-SCH	BCCH	FFFF

4.4 Experimental Setups

This section describes how to set up devices able to catch SIB and paging messages.

4.4.1 Overview

eNodeBs continuously broadcasts unprotected SIB and paging messages to all surrounding UEs. Consequently, a USRP and free Software Defined Radio (SDR) software collect and decode these messages. Unlike the setup described in Section 3.3, the USRP is configured as a UE. Notably, OpenAirInterface has not implemented the page procedure yet [Bhe16]; consequently, the software srsLTE captures paging messages, while OpenAirInterface captures SIB messages. The following hardware and software were used to conduct the experiments:

- Desktop computer, Ubuntu 14.04 LTS, 3.19.0-031900-lowlatency kernel
  - Memory 8 GiB
  - Processor Inter Core i7 860 @ 2.80GHz * 8
  - Graphics GeForce GT 630
- OpenAirInterface used to decode SIBs messages
- srsLTE used to decode paging messages
- USRP B200mini-i, 70 MHz - 6 GHz frequency range, full duplex and USB 3.0 bus-powered
- Atom - text editor v1.10.2

4.4.2 srsLTE

srsLTE is an open source platform for building SDR UE and eNodeB, developed by Software Radio Systems (SRS) [Sof]. The software is implemented in C and compatible with the LTE Release 8 specification, defined by 3GPP [Sof, 3GP13]. srsLTE contains multiple prebuilt example applications based on UE and eNodeB specifications. Example applications implemented in this thesis are the following:

- **pdsch_ue:** Behaves as a UE and provide measurement data from commercial LTE cells. Additionally, it catches and decodes paging and SIB type 1 messages
- **cell_measurement:** Provides continuous signal strength measurements for a particular LTE cell
- **cell_search:** Provides a list of neighboring LTE cells and their respective frequency

Compared to OpenAirInterface, srsLTE requires less disk space, fewer external dependencies and lower requirements for PC specifications. SrsLTE is intuitive and provides a satisfying user experience; however, OpenAirInterface provides a clearer separation between network entities. Section 3.3.2 describes the specifications of OpenAirInterface. Furthermore, `srsLTE's` Github repository contains all the open-source SRS projects.

4.4.3 Topology

USRP B200mini and the srsLTE software constitutes the topology of the Paging Catcher. The srsLTE software consists of three prebuilt applications: *pdsch_ue*,

cell_measurement, and *cell_search*. Figure 4.3 depicts the full topology of the Paging Catcher.

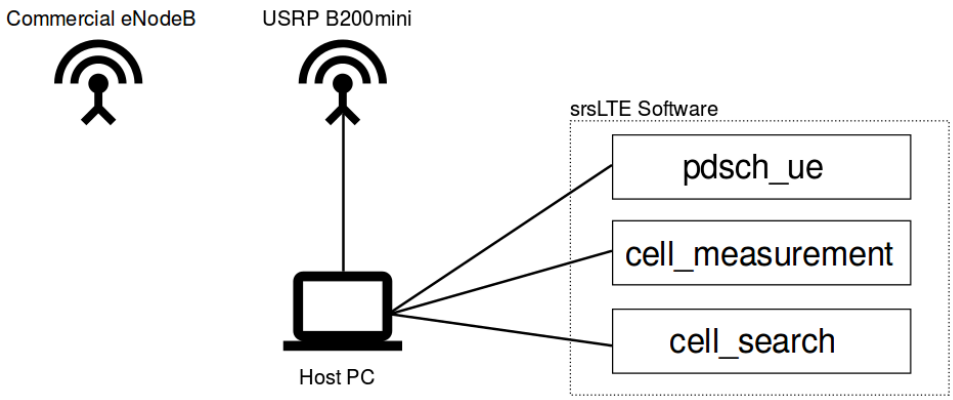

Figure 4.3: Topology of the Paging Catcher.

The topology of the SIB Catcher is very similar to the topology described in Section 3.3.4; however, the UE software replaces the EPC software. Moreover, the USRP B200mini collects broadcast data transmitted over the air by commercial eNodeBs. Figure 4.4 depicts the topology of the SIB Catcher.

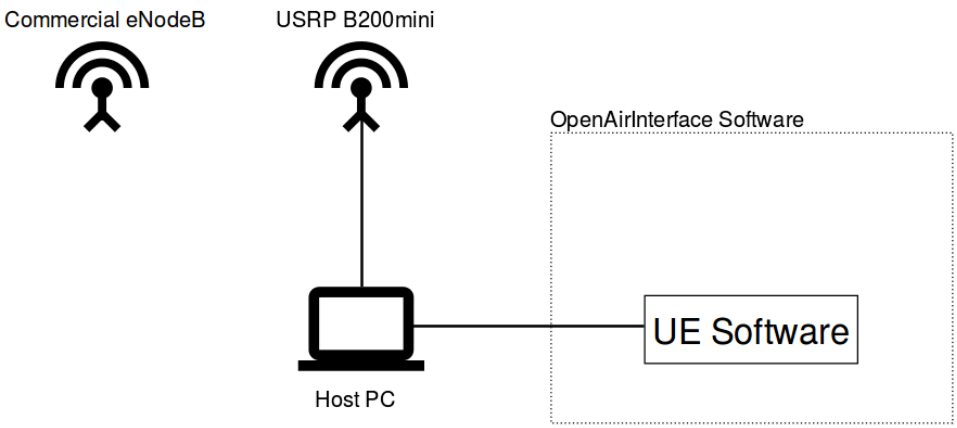

Figure 4.4: Topology of the SIB Catcher.

4.4.4 Using srsLTE as a Paging Catcher

SrsLTE and a USRP are needed to catch paging messages sent by commercial eNodeBs. The USRP is of the type B200min, and srsLTE is configured as a UE. Section 3.3.3 describes the full specification of the B200mini. Furthermore, srsLTE is easily downloaded and installed using the following command in the terminal[3]:

```
$ git clone https://github.com/srsLTE/srsLTE.git
$ cd srsLTE
$ mkdir build
$ cd build
$ cmake ../
$ make
```

Alternatively, the software suite can also be installed using the following command in the terminal [srs]:

```
$ sudo make install
```

SrsLTE automatically includes the prebuilt example applications during the installation process. Consequently, the prebuilt application *pdsch_ue* catches and decodes paging messages broadcasted from a commercial cell. Run the following commands to start *pdsch_ue*:

```
$ cd srsLTE/build/srsLTE/examples
$ ./pdsch_ue -f 806000000 -r fffe
```

Option -f represents the downlink frequency in GHz, while -r represents the RNTI. To determine the downlink frequency of a particular cell, 'finnsenderen.no' find the geographical location and the frequency band [Nko]. Figure 4.5 illustrates how 'finnsenderen.no' locates the target cell at the "Central Building 1" and the corresponding frequency band to be 800MHz.

[3]Mandatory dependencies: libfftw (library for computing the discrete Fourier transform).

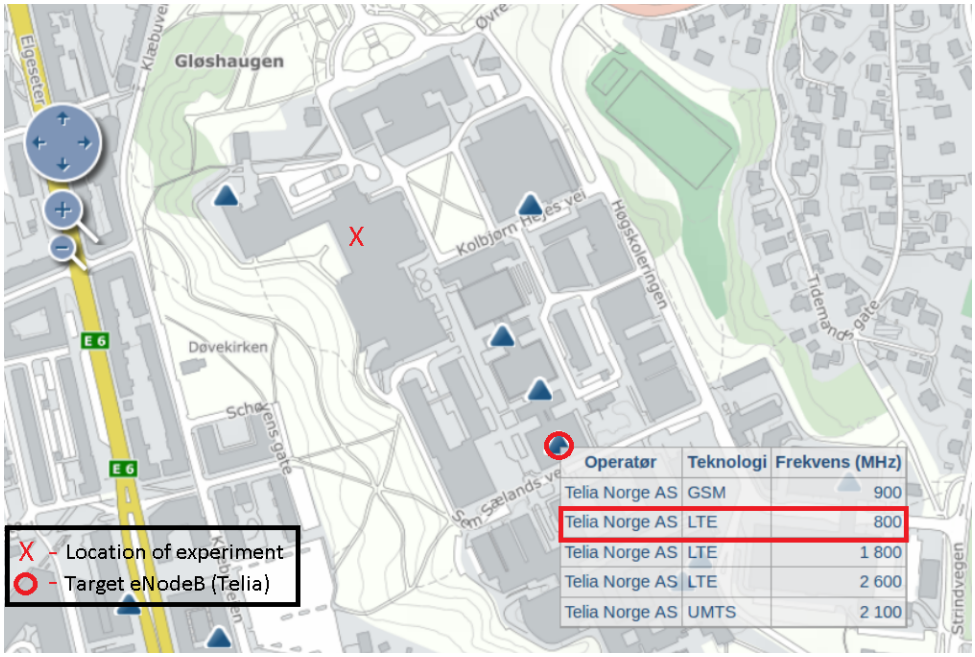

Figure 4.5: Overview of the neighboring LTE eNodeBs to the experiment location. The red "X" represents the location of the experiment, and the red "O" represents the location of the target cell. Edited map from 'www.finnsenderen.no' [Nko].

Subsequently, the prebuilt application *cell_search* determines the particular downlink frequency based on the frequency band. From Figure 4.5 one can observe that the target cell is operating in the 800MHz frequency band (band 20). Consequently, *cell_search* determines the downlink frequency of the neighboring cells operating in frequency band 20. Figure 4.6 illustrates the results gathered from *cell_search*.

```
Found 3 cells
Found CELL 806.0 MHz, EARFCN=6300, PHYID=123, 50 PRB, 2 ports, PSS power=-30.7 dBm
Found CELL 816.0 MHz, EARFCN=6400, PHYID=243, 50 PRB, 2 ports, PSS power=-5.4 dBm
Found CELL 816.0 MHz, EARFCN=6400, PHYID=112, 50 PRB, 2 ports, PSS power=-15.2 dBm
```

Figure 4.6: Surrounding cells in band 20, gathered from srsLTE.

Figure 4.6 show that the *cell_search* application was able to detect three cells in band 20. Moreover, the target cell has the downlink frequency 806MHz, EARFCN

number 6300 and physical identity 123. Consequently, all the required parameters for the *pdsch_ue* application are obtained.

Results

The results obtained in this section demonstrates how to gather paging messages, while Section 4.5 analyzes the content of the messages. Figure 4.7 illustrates how the Paging Catcher collects and decodes paging messages broadcasted by cell id 123. The Paging Catcher was sniffing the cell for two seconds and gathered eight paging messages.

```
Setting sampling rate 11.52 MHz
 - Cell ID:          123
 - Nof ports:        2
 - CP:               Normal
 - PRB:              50
 - PHICH Length:     Normal
 - PHICH Resources:  1
 - SFN:              228
Decoded MIBB. SFN: 228, offset: 3
[40 04 0e 0f fc 85 a8 00 00 ];
[40 03 8c 05 7f e0 30 00 00 ];
[40 03 8c 03 9f 04 38 00 00 ];
[40 03 8d 03 f7 39 00 00 00 ];
[40 03 8f 00 ea 84 18 00 00 ];
[40 03 8d 0a 43 a6 00 00 00 ];
[40 03 8f 08 59 cf 68 00 00 ];
[40 03 8d 0a 66 ac 10 00 00 ];
```

Figure 4.7: Paging messages from Cell ID 123, gathered from srsLTE.

Paging decoding is a two-step procedure: the first step catches and decodes paging message from PDSCH to raw Abstract Syntax Notation One (ASN.1) hexadecimal format (depicted in Figure 4.7), the second step decodes the raw data to readable Extensible Markup Language (XML) format (depicted in Figure 4.8). The code given in Appendix E shows how to decode paging messages in practice[4].

[4] "www.marben-products.com" is an online ASN.1 decoder which does not require any software [Mar].

```
<PCCH-Message>
  <message>
    <c1>
      <paging>
        <pagingRecordList>
          <PagingRecord>
            <ue-Identity>
              <s-TMSI>
                <mmec>01000000</mmec>
                <m-TMSI>11100000111111111100100001011010</m-TMSI>
              </s-TMSI>
            </ue-Identity>
            <cn-Domain>
              <cs/>
            </cn-Domain>
          </PagingRecord>
        </pagingRecordList>
      </paging>
    </c1>
  </message>
</PCCH-Message>
```

Figure 4.8: Decoded ASN.1 paging message.

4.4.5 Using OpenAirInterface as a System Information Catcher

The construction of a SIB Catcher requires OpenAirInterface and a USRP B200mini. Since commercial eNodeBs sends the SIB messages to UEs, OpenAirInterface must be configured as a UE. Furthermore, the OpenAirInterface-UE setup is much simpler in terms of complexity and needed entities. Whereas the setup described in Section 3.5 requires interaction between the UE and the EPC, this setup only receives messages from the eNodeB. The following command downloads the OpenAirInterface-UE project:

```
$ git clone https://gitlab.eurecom.fr/
            oai1B/openairinterface5g.git
```

Appendix A.1 and Appendix A.4 contains installation instructions and required dependencies for the OpenAirInterface-UE project. Subsequently, the following command initiates the SIB Catcher:

```
$ cd /oai1b/openairinterface5g/cmake_targets
$ sudo -E ./lte_noS1_build_oai/build/lte-softmodem-nos1
  -U -C806000100 -r50 --ue-scan-carrier --ue-txgain 90
  --ue-rxgain 115 -S -V -K /tmp/eNB0.log;ENABLE_ITTI=1 >&1
```

Option -C represents the downlink frequency of the target eNodeB, *–ue-txgain* sets the UE transmission gain, and *–ue-rxgain* sets the UE reception gain. The remaining parameters should always be included as they provide logs used to analyze the results. Moreover, the approach in Section 4.4.4 describes how to determine the parameter values for the target cell.

Results

OpenAirInterface-UE catches, decodes, and outputs the result in XML format. Figure 4.9 provides an excerpt of the gathered SIB messages, while Appendix F provides the complete output of the decoded SIB messages. As described in Section 4.3.3, SIB type 1 is scheduled once every 80th millisecond. From the results gathered in Appendix F, one can observe that SIB type 2 and 3 are scheduled once every 80th millisecond and SIB type 3-7 are scheduled once every 640th millisecond. Hence, the SIB Catcher only had to run for a few second to catch and decode all SIB messages[5].

The results given in Appendix F shows that catching SIB type 1-7 from commercial eNodeBs is feasible.

```
[RRC][I][decode_SIB1] [UE 0] : Dumping SIB 1
[RRC][I][decode_SI] [UE 0] Frame 616 Found SIB2 from eNB 0
[RRC][I][decode_SI] [UE 0] Received SIB1/SIB2/SIB3 Switching to RRC_SI_RECEIVED
[RRC][I][decode_SI] [UE 0] Frame 616 Found SIB3 from eNB 0
[RRC][I][decode_SI] [UE 0] Frame 641 Found SIB4 from eNB 0
[RRC][I][decode_SI] [UE 0] Frame 642 Found SIB5 from eNB 0
[RRC][I][decode_SI] [UE 0] Frame 643 Found SIB6 from eNB 0
[RRC][I][decode_SI] [UE 0] Frame 644 Found SIB7 from eNB 0
[RRC][I][decode_SI] SIStatus 7f, SIcnt 5/5
```

Figure 4.9: SIB messages gathered by the SIB Catcher.

4.5 Paging Analysis of Commercial PLMNs in Norway

This section analyzes the data gathered in Section 4.4.4 and explains how the data can be used to disclose the location of a subscriber.

4.5.1 Overview

As described in Section 4.2.1, LTE supports the smart paging feature which limits the paging to a cell. Consequently, this implies that the position of a subscriber can be mapped to the coverage area of a particular cell, which is typically 2 km^2 in

[5] How often UEs interprets SIB messages depends on the configurations of the UE.

urban areas. Moreover, this feature is not exploitable in GSM networks as the paging messages are sent to an entire location area (typically 100 km²) [Kun12]. Table 4.3 provides an overview of all the collected paging messages sorted by message type.

Table 4.3: Collected paging messages, sorted by message type.

PLMN	PagingRecord	systemInfoModification	systemInfoModification with ue-Identity	etws-Indication
Telia	2770	0	0	0
Telenor	92575	283	101	0
ice.net	1246	0	0	0

From Table 4.3 one can observe that most of the collected paging messages in this experiment are *PagingRecord* messages. Notably, 384 Telenor messages had the *systemInfoModification* indication, whereas 101 of these messages were destined for a specific subscriber. The *systemInfoModification* indicates that the core network has made some modifications to the BCCH [3GP16b]. Expectedly, none of the messages had the *etws-Indication*.

4.5.2 Using Social Media for Subscriber Mapping

Most of the paging messages include the *ue-Identity* field, which uniquely identifies a subscriber. Commonly, the *ue-Identity* contain the temporary identity S-TMSI. Consequently, mapping temporary identities to social identities is needed.

Previous Research

Previous research has discovered several techniques for mapping the temporary identity and the social identity; however, none of them are as effective anymore. Kune et al. proposed to initiate a phone call, and abort before the first ring [Kun12]. The phone call triggers paging but terminates before it displays on the UE. However, applications for detecting such activity have emerged [Uda]. Furthermore, Shaik et al. proposed to use Facebook Messenger to send a message to a person who is not in the friend list [SBA+15]. The message triggers paging but does not appear in the regular inbox folder. However, Facebook has changed this feature; users now see messages from persons not in the friend list as "Message Requests" [Fac].

Facebook Messenger Experiment

This experiment is inspired by the Facebook technique proposed by Shaik et al. [SBA+15]. The technique introduced by Shaik et al. is considered applicable; however, the technique described in this experiment is considered harder to detect and

hence improved. Notably, this attack will only work if the victim has the Facebook application installed on his/her LTE device.

This technique exploits the *"Filtered Requests"* feature in Facebook's messaging system. The *"Filtered Requests"* feature prevents spam and other unwanted messages from persons who are not in your friend list [Fac]. The *"Filtered Requests"* menu is buried under four menus, making it hard to locate and most people do not even know it exists[6]. Facebook have not publicly published which messages maps to the *"Message Requests"* and the *"Filtered Requests"*. However, experiments conducted in this thesis have revealed that messages from persons previously marked as spammers, maps to the "Filtered Requests" folder. Figure 4.10 depicts the buried location for the *"Filtered Requests"* folder.

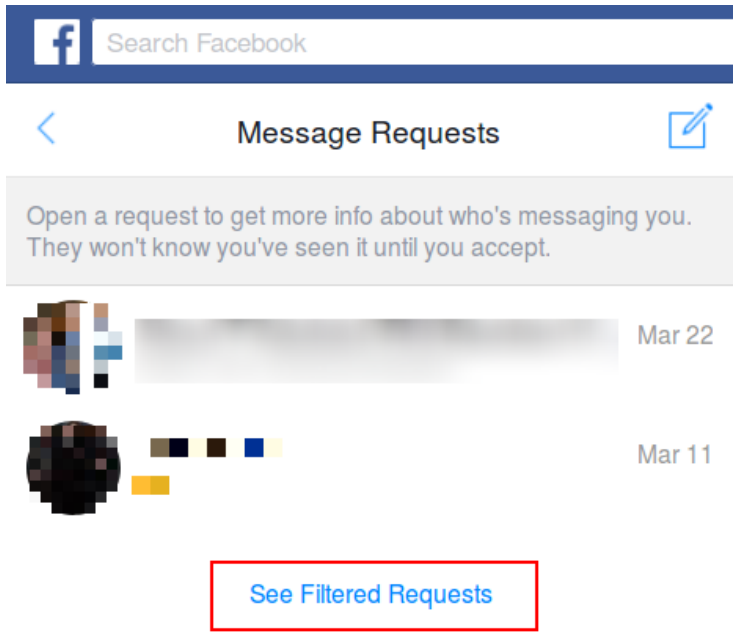

Figure 4.10: The hidden *"Filtered Requests"* feature in Facebook's messaging system.

[6]To get to the *"Filtered Requests"* menu click Messenger, then Settings, then Message Requests, and click "See Filtered Requests".

Two Facebook accounts are required to exploit the *"Filtered Requests"* feature: account1 used to send Facebook messages (paging messages), and account2 used to mark account 1 as a spammer. After account1 has been marked as a spammer, all subsequent messages to users not in the friend list redirects to the "Filtered Requests" folder. As a result, account1 can send paging messages to a specific subscriber with a low probability of the subscriber noticing the messages.

Figure 4.11 shows that by triggering five consecutive paging messages to the same subscriber, a mapping between the GUTI and the subscriber is determined. The figure highlights that the hex string "40 02 9c 20 ac 18 d0" is the only paging message repeated five times, and hence contains the GUTI of the subscriber.

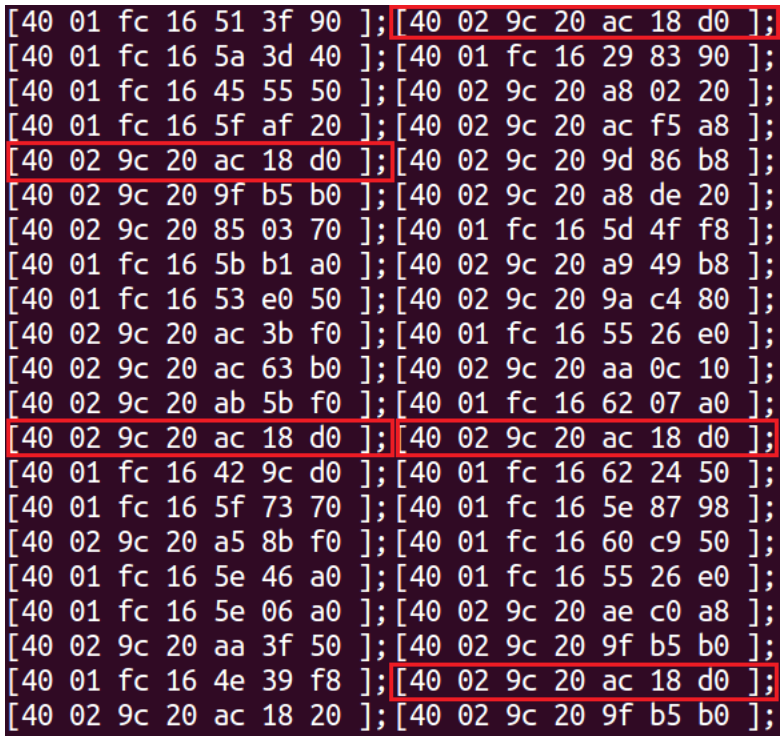

Figure 4.11: Five consecutive paging messages maps the GUTI to subscriber's social identity.

4.6 System Information Analysis of Commercial PLMNs in Norway

This section analyzes the SIB messages broadcasted by the three PLMN in Norway.

4.6.1 Overview

Three cells are analyzed in this experiment, one from each PLMN. All the analyzed cells operated in the frequency band 20 and were located relatively close together to make the conditions similar for each PLMN.

4.6.2 Telia

Table 4.4 highlights the most relevant parameters for this thesis, extracted from SIB 1-7 broadcasted by a nearby Telia eNodeB at the location of the experiment. Cell 34767628, transmitting on EARFCN 6300, was used during this experiment. Section 4.4.4 describes how to locate neighboring eNodeBs and corresponding EARFCNs. Moreover, Appendix F contains all the SIB 1-7 parameters broadcasted by cell 34767628.

Table 4.4: System information broadcasted by Telia eNodeB.

Parameter	**Value**	
MNC	242 (Norway)	
MNC	02 (Telia)	
TAC	0000100100000001 (2305)	
Cell Identity	00100001001010000001100001100 (34767628)	
Min RX Level	-66	
Frequency Band	20	
Scheduling Info	SIB 2-3 = 80 ms	SIB 4-7 = 640 ms
Default Paging Cycle	128 radio frames	
Cell Reselection Priority	5	
Intra Neighbors Cell List	Physical_Cell_Id=124, 200, 225, 226, 232, 231	
Inter Carrier Frequency Info	Cell_ID=123, DL_Frequency=2850, Reselection_Priority=7	Cell_ID=(47, 125, 123) DL_Frequency=1650, Reselection_Priority=6

Table 4.4 shows that all the required parameters for determining the inter-frequency cell reselection are given. Notably, the reselection priority of the sniffed

cell is 5 (Cell ID 34767628), while the reselection candidates, EARFCN 2850 and EARFCN 1650, have the priority 7 and 6 respectively[7]. Consequently, the sniffed cell operates with the lowest prioritized frequency and least likely to be selected.

4.6.3 Telenor

Table 4.5 highlights some of the parameters broadcasted by a nearby Telenor eNodeB at the location of the experiment. Cell 17803267, transmitting on EARFCN 6400, was used during this experiment. Section 4.4.4 describes how to locate neighboring eNodeBs and corresponding EARFCNs.

Table 4.5: System information broadcasted by Telenor eNodeB.

Parameter	**Value**
MNC	242 (Norway)
MNC	01 (Telenor)
TAC	0111011011000001 (30401)
Cell Identity	00010000111110101000000000011 (17803267)
Min RX Level	-64
Frequency Band	20
Scheduling Info	SIB 2-3 = 160 ms, SIB 5 = 320 ms, SIB 6-7 = 640 ms
Default Paging Cycle	128 radio frames
Cell Reselection Priority	5
Inter Carrier Frequency Info	(DL_Frequency=1450, Reselection_Priority=6), (DL_Frequency=3050, Reselection_Priority=6), (DL_Frequency=3248, Reselection_Priority=6), (DL_Frequency=251, Reselection_Priority=5), (DL_Frequency=1306, Reselection_Priority=5), (DL_Frequency=3750, Reselection_Priority=5),

[7]The cell reselection priority is represented as an integer between 0 and 7, where 7 is the highest priority possible [3GP16b].

Unexpectedly, no SIB type 4 message were captured during the experiment; there were also no SIB type 4 entries in the *schedulingInfoList*. Furthermore, the Telenor cell has a greater collection of reselection candidates, most likely due to that fact that Telenor has the largest number of eNodeBs in the given area and hence provides a broader frequency spectrum.

4.6.4 ice.net

Table 4.6 highlights some of the parameters broadcasted by a nearby ice.net eNodeB at the location of the experiment. Cell 64028178, transmitting on EARFCN 6200, was used during this experiment. Section 4.4.4 describes how to locate neighboring eNodeBs and corresponding EARFCNs.

Table 4.6: System information broadcasted by ice.net eNodeB.

Parameter	**Value**	
MNC	242 (Norway)	
MNC	14 (ice.net)	
TAC	1111001000110010 (62002)	
Cell Identity	00111101000011111111000010010 (64028178)	
Min RX Level	-60	
Frequency Band	20	
Scheduling Info	SIB 2-7 = 160 ms	
Default Paging Cycle	128 radio frames	
Cell Reselection Priority	5	
Intra Neighbors Cell List	Physical_Cell_Id=124, 200, 225, 226, 232, 231	
Inter Carrier Frequency Info	Cell_ID=123, DL_Frequency=2850, Reselection_Priority=7	Cell_ID=(47, 125, 123) DL_Frequency=1650, Reselection_Priority=6

Table 4.6 shows that the all the SIB messages are transmitted once every 160th millisecond. Unlike Telia which transmits SIB 2-3 once every 80th millisecond and SIB 4-7 once every 640th millisecond. Regarding efficiency, one may say that Telia's

configuration is better, as the most important SIBs are transmitted more often than the less important SIBs. Noteworthy, the reselection candidates for the ice.net cell is equivalent to the Telia cell, because ice.net and Telia have an agreement that allows ice.net subscribers to use the Telia infrastructure [Ice].

4.7 Paging Identity Analysis

This section analyses the usage of the paging parameter *ue-Identity*. As described in Section 4.2.3, *ue-Identity* identifies the paging message and should preferably contain the temporary identity of the subscriber.

4.7.1 Results and Discussion

Each PLMN were sniffed for one hour, and the results are summarized in Table 4.7. The paging messages were collected from Telia's cell 34767628, Telenor's cell 17803267 and ice.net's cell 64028178. The table includes Reference Signal Received Power (RSRP) to give an indication on the total received power for each cell[8].

Table 4.7: Paging statistics for Telia, Telenor, and ice.net.

PLMN	Number of Paging Messages	Number of IMSIs	Number of GUTIs	RSRP (dBm)
Telia	2770	0	2770	-25,3
Telenor	92959	0	92959	-18,8
ice.net	1246	0	1246	-30,3

Table 4.7 shows that the total number of received paging messages varies considerably among the three PLMNs. The reason for the uneven paging distribution is unknown; however, there seems to be a correlation between RSRP value and the number of received paging messages, which may indicate that some of the paging messages from Telia and ice.net are lost due to weaker received signal power. Another potential reason is that Telia and ice.net have implemented smart paging, which drastically reduces the total amount of transmitted paging messages [Dav13]. However, this is just speculations without any scientific proofs.

Expectedly, no IMSIs were found during the experiment, and the PLMNs have successfully preserved the confidentiality of the permanent subscriber identity.

[8]The higher the RSRP value, the stronger the received signal.

4.7.2 ice.net GUTI Persistence

Shaik et al. discovered that some network operators tend to not change the GUTI regularly [SBA+15]. They found that a moving UE was using the same GUTI for three days, exposing the UE for movement tracking based on the GUTI. Consequently, the motivation behind this experiment is to check the GUTI update interval for Norwegian network operators.

The GUTI persistence for ice.net has been monitored for six days, and the results are summarized below:

- The UE was moved 500 meters in an urban area, the movement caused a handover, and the UE went from frequency band 20 to frequency band 3. Consequently, the movement triggered a GUTI reallocation.
- The UE was camping in the same cell for 24 hours, during this period the S-TMSI changed several times. The experiment captured the following S-TMSIs during this period: 3255484813, 3244864778, and 3254888352.
- If the UE was completely turned off, a new GUTI was allocated when it was turned on.

Based on the observations above, the GUTI appears to change regularly. Although the experiment revealed that the first octet was similar for three consecutive S-TMSIs, there is no contiguous pattern for the subsequent octets; hence, the S-TMSIs are chosen randomly.

4.8 Countermeasures

The simplest method to countermeasure the attacks discussed in this chapter is to update the GUTI more frequent. This method requires no changes to the LTE protocol and prevents movement tracking [SBA+15].

4.9 Discussion and Results

All the experiments conducted in this chapter were executed in the wireless security lab at NTNU. The experiments have proven that paging messages broadcasted by commercial PLMNs can be captured and exploited to disclose the location of a subscriber. Moreover, the smart paging feature implemented in LTE networks allows an attacker to passively determine the location of a subscriber within a 2 km^2 area.

The paging procedure can also be actively exploited to determine the location of a subscriber. As described in Section 4.2, events such as SMS, voice calls, and Facebook notifications triggers the paging procedure. Consequently, a "hidden" Facebook message triggers a paging message, which can be sniffed by an attacker.

The srsLTE Paging Catcher is considered easy to implement. Minimal C programming skills are required, and no knowledge of the LTE protocol stack is needed. As a result, anyone with a USRP is capable of performing passive location disclosure attacks in LTE. Moreover, passive location disclosure attacks are hard to detect because they do not leave footprints in the network. Consequently, anyone in possession of a USRP and a PC manages to exploit the paging procedure implemented in commercial LTE networks.

Section 4.4.5 shows that commercial eNodeBs regularly broadcasts SIB type 1-7. OpenAirInterface-UE manages to catch and decode the SIB messages. Moreover, the SIB messages reveal valuable configuration information about the network, which can be exploited by an attacker in the information gathering phase/black-box testing. For the experiments conducted in this thesis, the SIB messages are actively used to exploit vulnerabilities in LTE networks.

Table 4.8 shows a summary of all the paging messages gathered in this chapter. In total, 96975 paging messages were captured in an hour; unexpectedly, 99,96% of the messages originated from a Telenor cell. A possible explanation for the unbalanced signaling load might be that Telenor has neglected to implement smart paging. Nowoswiat suggested that smart paging reduces the signaling by as much as 80% [Dav13]. Consequently, by reducing Telenor's number of paging message by 80%, the total number of paging messages would be much closer to Telia and ice.net.

Table 4.8: Summary of all the gathered paging messages.

	Telia	Telenor	ice.net
Number of Paging Messages	2770	92959	1246
systemInfoModification	0	384	0
pagingRecord	2770	92575	1246
etws-Indication	0	0	0
GUTI	2770	92959	1246
IMSI	0	0	0
Multiple Identities	125	13173	23

Furthermore, Table 4.3 categorizes the gathered paging messages. 384 collected Telenor messages contain the *SystemInfoModification*, indicating that the network

has made changes to the BCCH. Table 4.7 shows that none of the 96975 captured paging messages include the permanent identity of the subscriber. As a result, all the Norwegian PLMNs manages to keep the IMSI undisclosed and maintain the confidentiality of the permanent subscriber identity.

The test scenarios described in Section 4.7.2 were implemented to evaluate the GUTI persistence for ice.net. Notably, the results gathered in Section 4.7.2 build on the observations in [SBA+15]. Expectedly, the results revealed that ice.net manages to update the GUTI regularly, and follows the GUTI reallocation recommendations by 3GPP [3GP11c].

Chapter 5
Existing Location Disclosure Attacks

This chapter explains existing location disclosure attacks in LTE networks. Two active location disclosure attacks are technically explained and analyzed. Additionally, this chapter contains improvements for the proposed attacks.

Chapter 2 and Chapter 3 provide relevant information for understanding the underlying expressions used in this chapter. Section 2.6 includes a description of the attach procedure and the message exchange between the UE and the network, while Section 3.4 explains how to force a UE to attach to an IMSI Catcher and retrieve the IMSI.

Shaik et al. proposed two active location disclosure attacks: via measurement reports and Radio Link Failure (RLF) reports [SBA+15]. Both attacks heavily rely on IMSI Catchers and exploits specification and implementation vulnerabilities in LTE. Furthermore, the two attacks locate subscribers with high accuracy.

5.1 Measurement Report

The measurement report is a result of cell measurements performed by the UE. The UE performs measurements both in IDLE state and in CONNECTED state [Shaa]. SIB messages determine the IDLE state measurements used for cell selection/reselection, while specific RRC messages determine CONNECTED state measurements used for eNodeB handover [Shaa]. The following measurement report attack exploits the CONNECTED state measurements.

5.1.1 Trigger and Obtain Measurement Report

To trigger a measurement report from a particular UE, an IMSI Catcher is required. The attacker forces the UE to connect to the IMSI Catcher by applying the steps described in Section 3.4. Subsequently, the UE completes the RRC connection procedure and enters into CONNECTED state, as illustrated in Figure 5.1. Next, the

IMSI Catcher constructs an *RRCConnectionReconfiguration* message with different cell ID and sends it to the UE [SBA+15]. Notably, to receive an unencrypted measurement report in return, the IMSI Catcher send the *RRCConnectionReconfiguration* unprotected. Upon reception of the *RRCConnectionReconfiguration* message, the UE computes signal power from neighboring cells and sends the result in an unprotected *measurementReport*. Figure 5.1 illustrates the complete message exchange between the UE and the eNodeB (IMSI Catcher).

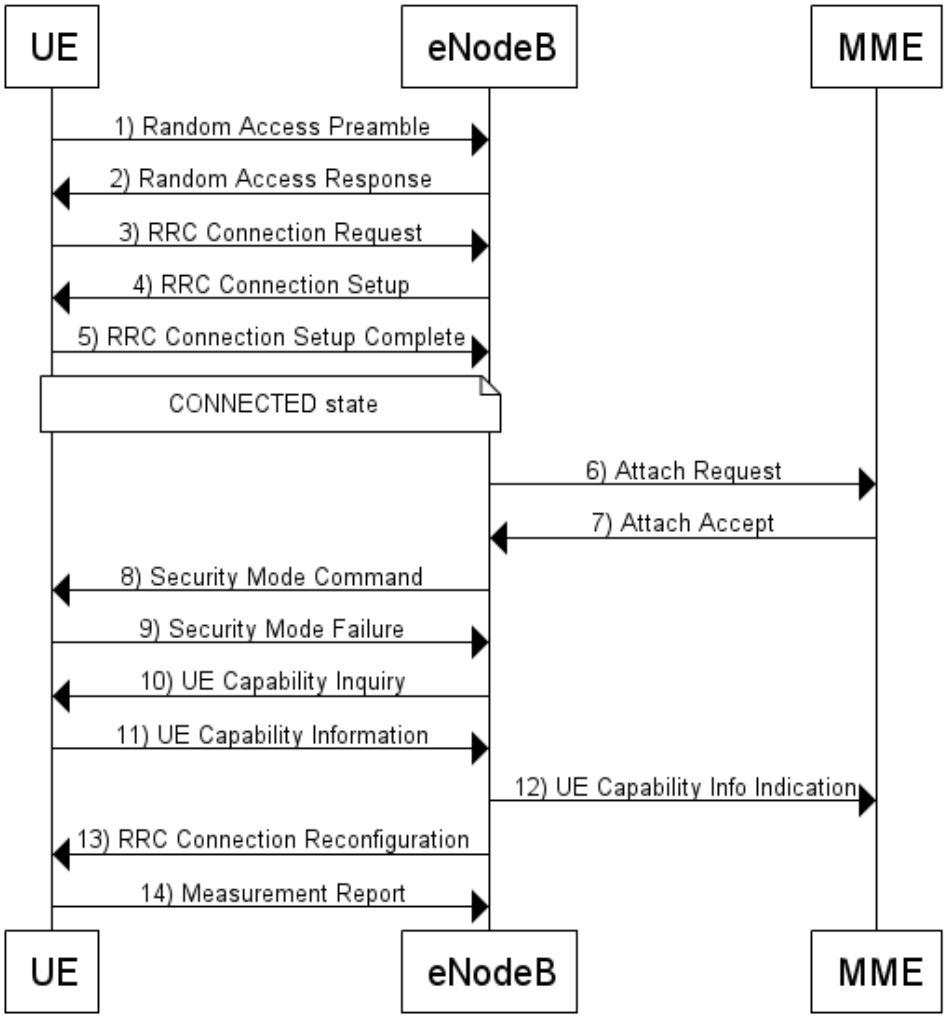

Figure 5.1: Retrieving measurement report from UE.

As illustrated in Figure 5.1, the *SecurityModeCommand* message is sent unprotected, which trigger *SecurityModeFailure* by the UE and hence deactivates integrity and confidentiality protection [3GP16b].

```
<UL-DCCH-Message>
    <message>
        <c1>
            <measurementReport>
                <criticalExtensions>
                    <measurementReport-r8>
                        <measResults>
                            <measResultServCell>
                                <srspResult>
                                    68
                                </srspResult>
                                <srsqResult>
                                    21
                                </srsqResult>
                            <measResultServCell>
                            <measResultNeighCells>
                                <measResultListEUTRA>
                                    <measResultEUTR>
                                        <physCellId>
                                            123
                                        </physCellId>
                                        <measResult>
                                            <rsrpResult>
                                                69
                                            </rsrpResult>
                                        </measResult>
                                    </measResultEUTR>
                                    <measResultEUTR>
                                        <physCellId>
                                            125
                                        </physCellId>
                                        <measResult>
                                            <rsrpResult>
                                                70
                                            </rsrpResult>
                                        </measResult>
                                    </measResultEUTR>
                                </measResultListEUTRA>
                            </measResultNeighCells>
                        </measResults>
                    </measurementReport-r8>
                </criticalExtensions>
            </measurementReport>
        </c1>
    </message>
</UL-DCCH-Message>
```

Figure 5.2: Structure of a measurement report message.

Figure 5.2 illustrates the structure of a *measurementReport* message[1]. The figure shows that the serving cell receives RSRP 68 and Reference Signal Received Quality (RSRQ) 21, while the neighboring cells 123 and 125 receives RSRP 69 and RSRP 70 respectively[2]. Consequently, by applying the measurement results in the trilateration process described in Section 5.3, a subscriber's exact location is determined. Notably, the trilateration process requires measurement data from three neighboring cells to calculate the position of a subscriber.

5.1.2 Measurement Report Improvements

The improvements portrayed in this section have not been practically implemented; however, the theory suggests that improvements are feasible.

The proposal combines the IMSI Catcher attack described in Section 3.5 with *measurementReport* acquisition. By combining IMSI catching and *measurementReport* acquisition into one attack, an attacker will be able to obtain the IMSI and determine the exact location of a subscriber.

Figure 5.3 illustrates the message flow of the improvement proposal. Message 1-5 represent the RRC connection procedure, whereas message 5 sends the *Attach Request* and triggers the attach procedure. Instead of forwarding the attach request to the MME, the eNodeB initiated an *Identity Request*, requesting the IMSI from the UE. Consequently, the UE attaches its IMSI to the *Identity Response* message and sends it to the eNodeB (IMSI Catcher). At this point, the IMSI Catcher have obtained the subscriber's IMSI. Subsequently, the security procedure and the connection reconfiguration completes as described in Section 5.1.1. As a result, the IMSI Catcher collects the IMSI and the measurement data in one operation. Finally, the IMSI Catcher terminates the connection by sending an *RRCConnectionRelease* message to the UE. The *RRCConnectionRelease* message with *releaseCause* "other" instructs the UE to return to IDLE state and attach to a new cell [3GP16b, 3GP16c]. The UE performs cell reselection based on the cell reselection priority broadcasted in the SIBs [3GP16b].

[1]The *measurementReport* depicted in Figure 5.2 does not contain real data captured from a live network. The content is produced based on the information in [Shaa].

[2]The 3GPP standard provides a mapping between RSRP/RSRQ values and measured quality value (dBm) [3GP13].

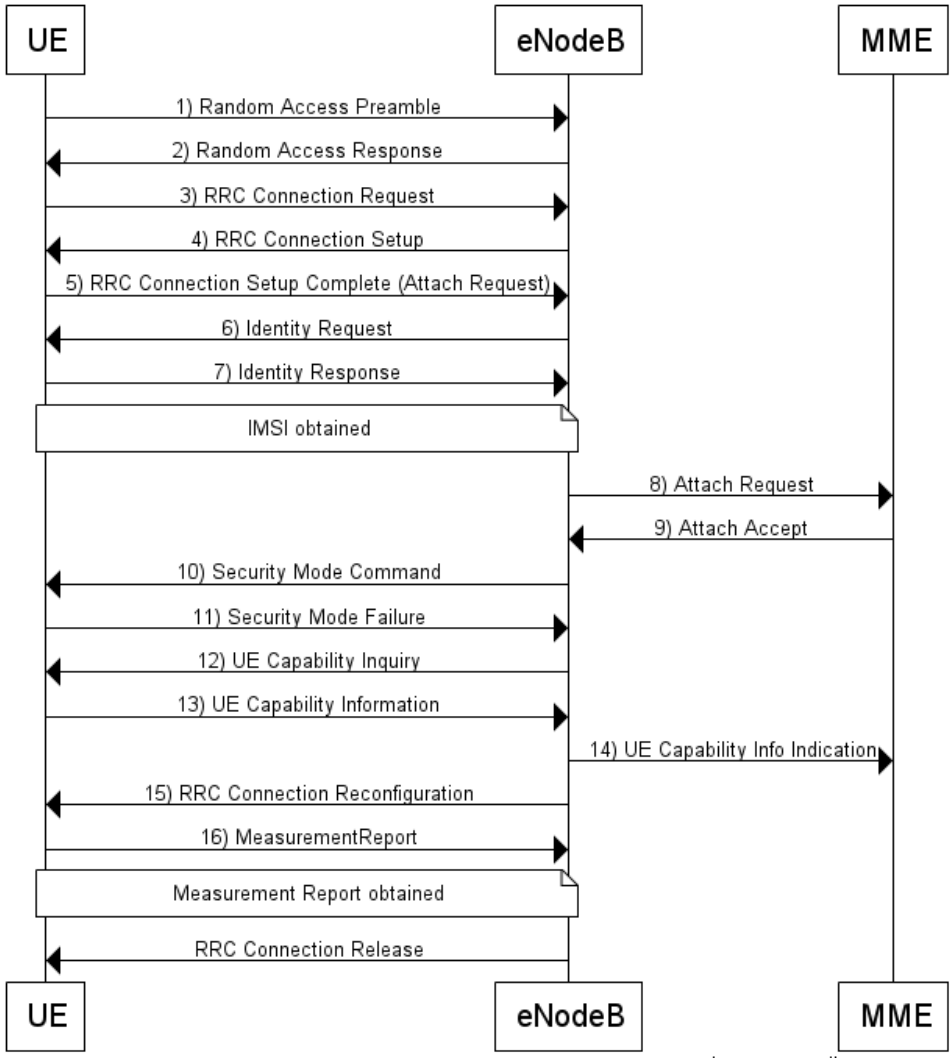

Figure 5.3: Combined measurement report and IMSI acquisition.

5.2 RFL Report

The *UEInformationRepsonse* message contains the RLF report, used to provide information requested by the eNodeB [3GP16b]. Moreover, the UE generates the RLF report because of poor radio conditions [Qua10].

5.2.1 RLF Report Structure

Table 5.1 contains an overview of all the possible fields in the RLF report.

Table 5.1: Content and structure of the RFL report [3GP16b].

Field	Description
measResultLastServCell	Provides the latest measurement results from the last serving cell, where the RLF occurred. The results are given in RSRP and RSRQ.
measResultNeighCells	Contains a list of measurement results of the best reported neighboring cells. The list includes cell ID, RSRP, and RSRQ for each cell.
locationInfo	Provides detailed location information about the UE, used to correlate measurements and UE position information [3GP16b].
failedPCellId	Indicates the identity of the cell in which the RLF occurred. The UE selects the same EARFCN used when the failure occurred.
reestablishmentCellId	Indicates the identity of the cell in which the re-establishment attempt was made.
timeConnFailure	Indicates the duration since last connection failure.
connectionFailureType	Indicates if the failure was due to RLF or handover failure.
previousPCellId	Identifies the source cell of the previous handover.

The content in the *UEInformationRepsonse* message depends on the parameters specified in the *UEInformationRequest* message, sent by the eNodeB.

5.2.2 Trigger and Obtain RLF Report

Two fake base stations are needed to obtain the RLF report: enb1, used to trigger a RLF; and enb2, used to collect the generated RLF report[3]. The attacker forces the UE to connect to enb1 by applying the steps described in Section 3.4, while enb2 is turned off. Further, enb1 immediately turns off as the UE enters into CONNECTED state, which causes an RLF scenario. Simultaneously enb2 turns on, as shown in Figure 5.4. When the UE detects the RLF caused by enb1, it starts the RLF timer (T310) [3GP16b]. Eventually, the UE creates the RLF report when T310 expires [3GP16b].

[3]Essentially, enb1 and enb2 are configured as IMSI Catchers.

Subsequently, the UE enters IDLE state and starts searching for other cells. The UE connects to enb2, enters into CONNECTED state and indicates the available RLF report in a *TAU Request* message. Consequently, enb2 request the RLF report by sending an unprotected *UEInformationRequest* message to the UE. As a result, the UE replies by attaching the RLF report to an unprotected *UEInformationResponse* message and sends it to enb2.

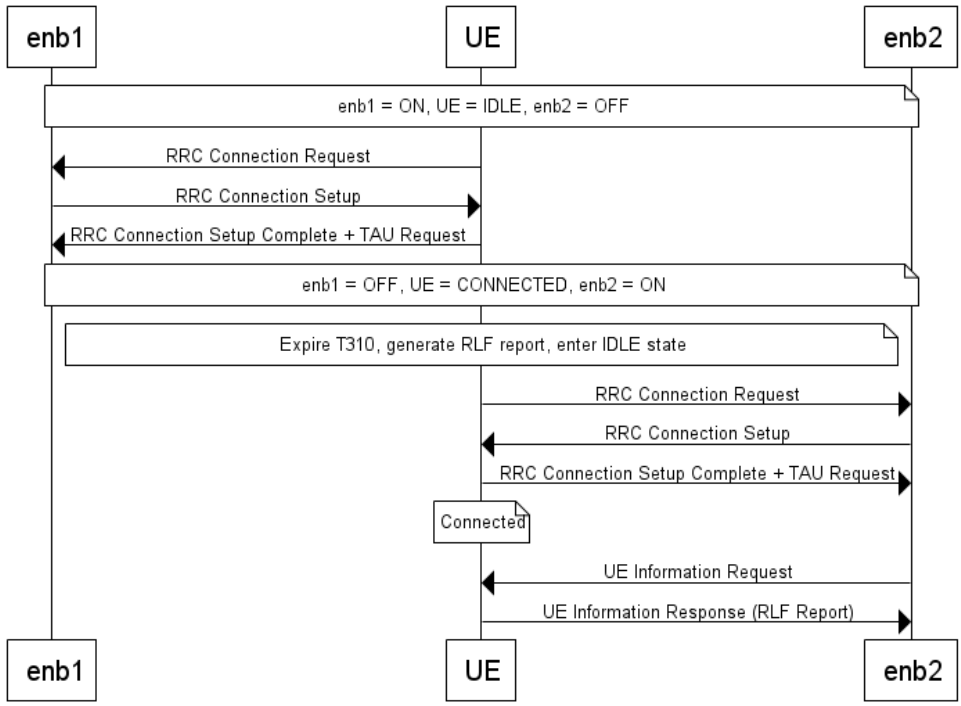

Figure 5.4: Acquiring the RLF report from UE.

The *measResultNeighCells* field in the RLF report contains measurement results from surrounding cells. Consequently, by using the trilateration process described in Section 5.3, the subscriber's exact location is determined.

5.2.3 RLF Report Improvements

The improvements portrayed in this section have not been practically implemented; however, the theory suggests that improvements are feasible. The following proposal improves the enb1 functionality, while enb2 functionality remains the same.

The Jammer functionality described in Section 3.4 inspired the improvements proposed in this section. Enb1 is configured as a Jammer, used to block the frequency of the serving cell. Hence, the cell becomes unavailable and disconnects the UE. Furthermore, the UE enters into IDLE state and starts searching for other cells. From this point forward, the functionality is the same as for the original attack: the UE attaches to enb2 and sends an unprotected *UEInformationRepsonse* message containing the RLF report.

The proposal reduces the total amount of messages exchanged between enb1/enb2 and the UE, and shortens the overall outage time for the UE. Additionally, the attack could be combined with the IMSI Catcher attack described in Section 3.5 to combine the IMSI and the RLF report acquisition into one process.

5.3 Determine Subscriber's Location Using Trilateration

Both the measurement report and the RLF report provides measurement data from neighboring cells, which are used by the trilateration process to determine a subscriber's approximate location. Figure 5.5 shows how the trilateration process combines measurement data from three neighboring cells to locate the subscriber. Moreover, the subscriber's approximate location lies within the intersection zone of the three cells, marked as a solid red area in Figure 5.5.

Figure 5.5: Locating a subscriber using the trilateration procedure. The solid red area indicates the location of the subscriber. Source: [SBA+15].

Furthermore, the process of calculating the distance estimate (d1, d2, and d3) based on the signal strength are described in [Caf98].

Notably, if the RLF report contains the *locationInfo* field, an attacker could determine the subscriber's exact location by using Global Positioning System (GPS) coordinates[4].

5.4 Discussion and Results

The research done by Shaik et al. forms the basis for the measurement report attack and RLF report attack [SBA+15]. Since Shaik et al. have discussed the results from the measurement report and the RLF report attack, the attacks have not been practically implemented in this thesis [SBA+15]. Furthermore, improvement proposals are elucidated to increase the efficiency of the attacks.

Both the measurement report attack and the RLF attack determine a subscriber exact position using the trilateration procedure. Moreover, the trilateration procedure requires measurement data from three commercial cell to locate a subscriber. Section 5.1.1 shows how an unprotected *RRCConnectionRequest* message triggers the measurement report, while Section 5.2.2 show how to exploit the RLF timer T310 to generate the RLF report.

From the proposal in Section 5.1.2, it can be observed that the measurement attack is extended to enrich the results. Figure 5.3 shows how the IMSI catching and the measurement report acquisition are combined into one operation. Furthermore, Section 5.2.3 illustrates how a Jammer simplifies the RLF report attack, and still obtain the same results. Notably, neither of the improvement proposals have been practically implemented; however, the theory suggests that improvements are feasible.

[4]Whether the *locationInfo* field is included in the RLF report or not, depends on the UE model [SBA+15].

Chapter 6
Conclusion

LTE IMSI Catchers and Paging Catchers have been studied in this thesis. A technical description of the subject explains how to use open source software for location disclosure and movement tracking. The IMSI Catcher and the Paging Catcher have been used to implement several attacks against privacy in LTE, the results of which are technically explained and analyzed.

Accordingly, an LTE IMSI Catcher has been implemented and successfully proven that IMSI catching in LTE networks is feasible. Moreover, since UEs connected to an LTE network does not necessarily perform handover based on the highest received signal power, the IMSI Catcher principles in GSM and UMTS are not applicable to LTE IMSI Catchers. Consequently, the LTE IMSI Catcher presented in this thesis exploits the *absolute priority* feature, implying that UEs attaches to the cells operating with high priority frequencies. Chapter 3 includes the actual IMSI catching experiment, illustrating real packet captures of the message exchange between the IMSI Catcher and the UE. The experiment revealed that IMSI catching in LTE is indeed achievable. Additionally, a thorough description of how to use the obtained IMSIs for determining the position of the subscribers are given.

Additionally, methods for obtaining subscriber identities passively have successfully been implemented. Paging Catchers acquires and decodes broadcast paging messages sent by commercial eNodeBs. The paging messages contain the temporary subscriber identity and were exploited in attacks against privacy in LTE. Although only temporary identities were revealed during the experiment, a mapping between social identity and temporary identity were achieved using Facebook Messenger. Moreover, the smart paging feature in LTE made it possible to locate a subscriber within a 2 km^2 area. As a result, both the identity and the location of the subscriber were revealed during the experiment. Notably, Chapter 4 included a practical paging identity experiment revealing that most of the collected paging messages originated from Telenor, which may indicate that Telenor does not implement smart paging. Additionally, Chapter 4 also describes how to catch SIB messages from commercial

BSs passively. SIB messages contain detailed information about the mobile operator and are utilized to configure the IMSI Catcher and the Paging Catcher.

Existing location disclosure attacks have been technically explained in Chapter 5. Shaik et al. have proposed two attacks against privacy in LTE [SBA+15]. Both attacks actively use an IMSI Catcher to accurately locate a subscriber using the trilateration technique. Improvements have been proposed to consolidate the existing attacks.

The attacks in this thesis have proven that LTE networks do no provide privacy of subscribers with regard to location and movement tracking. Both active and passive attacks have successfully demonstrated that LTE network operators in Norway do not preserve subscriber identity confidentiality.

6.1 Further Work

6.1.1 Implementation of LTE IMSI Catcher with Extended Coverage Area

The LTE IMSI Catcher with extended coverage area proposed in Section 3.6.2 suggests that by combining the coverage area of the previously attached cell and the coverage of the IMSI Catcher, the location accuracy of the subscriber is improved. However, this proposal is just a hypothesis without any guarantees that it will work in practice. Consequently, a practical implementation of the proposal could be used to determine the feasibility.

6.1.2 Smart Paging Analysis for Norwegian Operators

Section 4.5.1 explains how smart paging locates a subscriber within a 2 km^2 area. Although the smart paging technique locates a subscriber within a given area, there is no proof of Norwegian operators implementing the technique. Consequently, further work may include a practical experiment determining whether Norwegian mobile operators implements smart paging. Smart paging could be determined by observing if a particular paging message is sent to one cell only.

6.1.3 Implementation of Improvement Proposals

Section 5.1.2 and Section 5.2.3 theoretically explains how the measurement report attack and the RLF report attack could be improved. However, none of the improvement proposals have been practically implemented. Consequently, the efficiency of the proposals could be determined in further work. Notably, both proposals would require modifications to the source code; hence C experience is an advantage.

6.1.4 Countermeasures

Countermeasures for the IMSI Catcher and the Paging Catcher have only been discussed briefly in this thesis. In further work, more countermeasures should be proposed and carefully reviewed. Also, countermeasures should be verified in practical experiments.

References

[3GP] 3GPP. LTE. http://www.3gpp.org/technologies/keywords-acronyms/98-lte, [Online; Accesses 19.04.2017].

[3GP08a] 3GPP. 3G security; Security architecture. TS 33.102, 3rd Generation Partnership Project (3GPP), June 2008.

[3GP08b] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA) ; S1 Application Protocol (S1AP). TS 36.413, 3rd Generation Partnership Project (3GPP), September 2008.

[3GP08c] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation. TS 36.211, 3rd Generation Partnership Project (3GPP), September 2008.

[3GP08d] 3GPP. Network architecture. TS 23.002, 3rd Generation Partnership Project (3GPP), September 2008.

[3GP10] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN); Overall description; Stage 2. TS 36.300, 3rd Generation Partnership Project (3GPP), June 2010.

[3GP11a] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Repeater radio transmission and reception. TS 36.106, 3rd Generation Partnership Project (3GPP), January 2011.

[3GP11b] 3GPP. MME Related Interfaces Based on Diameter Protocol. TS 29.272, 3rd Generation Partnership Project (3GPP), December 2011.

[3GP11c] 3GPP. Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3. TS 24.301, 3rd Generation Partnership Project (3GPP), June 2011.

[3GP12a] 3GPP. 3GPP System Architecture Evolution (SAE); Security architecture. TS 33.401, 3rd Generation Partnership Project (3GPP), June 2012.

[3GP12b] 3GPP. Numbering, addressing and identification. TS 23.003, 3rd Generation Partnership Project (3GPP), March 2012.

[3GP13] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management. TS 36.133, 3rd Generation Partnership Project (3GPP), September 2013.

[3GP14a] 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), September 2014.

[3GP14b] 3GPP. Mobile radio interface layer 3 specification core network protocols; Stage 2 (structured procedures). TS 23.108, 3rd Generation Partnership Project (3GPP), September 2014.

[3GP14c] 3GPP. Mobile radio interface Layer 3 specification; Core network protocols; Stage 3. TS 24.008, 3rd Generation Partnership Project (3GPP), September 2014.

[3GP16a] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification. TS 36.321, 3rd Generation Partnership Project (3GPP), January 2016.

[3GP16b] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification. TS 36.331, 3rd Generation Partnership Project (3GPP), January 2016.

[3GP16c] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode. TS 36.304, 3rd Generation Partnership Project (3GPP), January 2016.

[Ari13] Arijit Satpathy . All about SIB's in LTE. http://lteinwireless.blogspot.no/2011/06/all-about-sibs-in-lte.html, [Online; Accesses 11.05.2017], 2013.

[Bhe16] Bheemarjuna Reddy Tamma, Rohit Gupta and Kiran Kuchi. OpenAirInterface (OAI) for Experimentation in 5G. http://www.iith.ac.in/newslab/sites/default/files/Documents/ANTStutorial.pdf, [Online; Accesses 08.05.2017], nov 2016.

[Bra89] Braden, Robert. RFC-1122: Requirements for internet hosts. *Request for Comments*, pages 356–363, 1989.

[Caf98] Caffery, James J and Stuber, Gordon L. Overview of radiolocation in CDMA cellular systems. *IEEE Communications Magazine*, 36(4):38–45, 1998.

[Can] Canonical Ltd (Ubuntu). https://www.ubuntu.com/, [Online; Accesses 04.05.2017].

[Cho10] Bong Youl (Brian) Cho. 3GPP LTE (Rel. 8) Overview. Technical report, Intel Corporation, September 2010.

[Cic16] Cichonski, Jeffrey and Franklin, Joshua M and Bartock, Michael. LTE Architecture Overview and Security Analysis. *NIST Draft NISTIR*, 8071, 2016.

[Cox12] Christopher Cox. *An introduction to LTE: LTE, LTE-advanced, SAE and 4G mobile communications*. John Wiley & Sons, 2012.

[Dav13] David Nowoswiat. Managing LTE Core Network Signaling Traffic. https://insight.nokia.com/managing-lte-core-network-signaling-traffic, [Online; Accesses 13.05.2017], jul 2013.

[Ett] Ettus Research. USRP Hardware Driver and USRP Manual. http://files.ettus.com/manual/page_usrp_b200.html, [Online; Accesses 03.05.2017].

[Fac] Facebook. Which messages will I get on Facebook? https://www.facebook.com/help/427500684120337?helpref=faq_content, [Online; Accesses 14.05.2017].

[Fed] Federal Communications Commission. CDBS Public Access. http://licensing.fcc.gov/prod/cdbs/pubacc/prod/cdbs_pa.htm, [Online; Accesses 07.06.2017].

[FHMN12] Dan Forsberg, Günther Horn, Wolf-Dietrich Moeller, and Valtteri Niemi. *LTE security*. John Wiley & Sons, 2012.

[Fre17] Frederic Firmin. The Evolved Packet Core. http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core, [Online; Accesses 25.04.2017], 2017. 3rd Generation Partnership Project (3GPP).

[Gio16] Giovanni Romano. 3GPP RAN progress on "5G". ftp://www.3gpp.org/Information/presentations/presentations_2016/3GPP%20RAN%20Progress%20on%205G%20-%20NetFutures.pdf, [Online; Accesses 08.06.2017], 2016.

[Goo] Google Inc. Google Earth Pro. https://www.google.com/intl/no/earth/desktop/, [Online; Accesses 25.05.2017].

[GSA15] GSA. Evolution to LTE report 4G MARKET & TECHNOLOGY UPDATE. Ts, Global mobile Suppliers Association, August 2015.

[Ice] Ice.net. Dekning. https://www.ice.no/private/coverage/, [Online; Accesses 16.05.2017].

[Int16] International Telecommunication Union. Mobile Network Codes (MNC) for the international identification plan for public networks and subscriptions (According to Recommendation ITU-T E.212 (09/2016)). Technical report, International Telecommunication Union, 2016.

[KG10] Ralf Kreher and Karsten Gaenger. *LTE signaling: troubleshooting and optimization*. John Wiley & Sons, 2010.

[Kun12] Kune, Denis Foo and Koelndorfer, John and Hopper, Nicholas and Kim, Yongdae. Location leaks on the GSM Air Interface. *ISOC NDSS (Feb 2012)*, 2012.

[Leu12] Leu, Fang-Yie and You, Ilsun and Huang, Yi-Li and Yim, Kangbin and Dai, Cheng-Ru. Improving security level of LTE authentication and key agreement procedure. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 1032–1036. IEEE, 2012.

[Lib] Libmich. What is libmich. https://github.com/mitshell/libmich, [Online; Accesses 06.06.2017].

[LJL+16] Marc Lichtman, Roger Piqueras Jover, Mina Labib, Raghunandan Rao, Vuk Marojevic, and Jeffrey H Reed. LTE/LTE-a jamming, spoofing, and sniffing: threat assessment and mitigation. *IEEE Communications Magazine*, 54(4):54–61, 2016.

[LLM+09] Anna Larmo, Magnus Lindström, Michael Meyer, Ghyslain Pelletier, Johan Torsner, and Henning Wiemann. The LTE link-layer design. *IEEE Communications magazine*, 47(4), 2009.

[Luc09] Lucent, Alcatel. The LTE network Architecture—A comprehensive tutorial. *Strategic Whitepaper*, 2009.

[Mar] Marben. Free Online 3GPP LTE ASN.1 Messages Decoder. http://www.marben-products.com/asn.1/services/decoder-asn1-lte.html, [Online; Accesses 12.05.2017].

[Mat] Matt Ettus. Ettus Research Update. http://static1.1.sqspcdn.com/static/f/679473/23654458/1381240753367/grcon13_ettus_products.pdf?token=ldHVQF0yAdZLWvdjhPjqLtrhB9I%3D, [Online; Accesses 03.05.2017].

[Mey04] Meyer, Ulrike and Wetzel, Susanne. A man-in-the-middle attack on UMTS. In *Proceedings of the 3rd ACM workshop on Wireless security*, pages 90–97. ACM, 2004.

[MO17] Stig F Mjølsnes and Ruxandra F Olimid. Easy 4G/LTE IMSI Catchers for Non-Programmers. *arXiv preprint arXiv:1702.04434*, 2017.

[Nas] Nasjonal Kommunikasjonsmyndighet. Spektrumtillatelser. http://frekvens.nkom.no/Frekvensportalen/tillatelser.xhtml, [Online; Accesses 04.04.2017].

[New] RCR Wireless News. Master LTE with the Help of an LTE Network Diagram. Available: http://www.rcrwireless.com/20140509/evolved-packet-core-epc/lte-network-diagram.

[Nit16] Nitin Gupta. Detail Explanation of MIB in LTE? http://www.sharetechnote.com/html/Paging_LTE.html, [Online; Accesses 10.05.2017], apr 2016.

[Nko] Nkom. http://finnsenderen.no/finnsender, [Online; Accesses 01.05.2017].

[NN03] Valtteri Niemi and Kaisa Nyberg. *UMTS security*. John Wiley & Sons, 2003.

[Nor06] Quinn Norton. GNU Radio Opens an Unseen World. Available: http://archive.wired.com/science/discoveries/news/2006/06/70933, May 2006.

[Ooi15] Ooi, Joseph. IMSI Catchers and Mobile Security. *School of Engineering and Applied Science University of Pennsylvania*, 2015.

[Opea] OpenAirInterface. OAI GitLab. https://gitlab.eurecom.fr/oai, [Online; Accesses 27.05.2017].

[Opeb] OpenBTS. Open Source Cellular Infrastructure. http://openbts.org/, [Online; Accesses 29.05.2017].

[Ope16] OpenAirInterface. Why is there a need of open source for 5G? Available: http://www.openairinterface.org/?page_id=72, [Online; Accesses 06.04.2017], 2016.

[Ope17] OpenAirInterface. How to Connect OAI eNB (USRP B210) with COTS UE. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/HowToConnectCOTSUEwithOAIeNBNew, [Online; Accesses 06.04.2017], 2017.

[Poo12] Poole, Ian. LTE physical, logical and transport channels. *Radio-Electronics. com*, 2012.

[Pro] Prof. Do van Thanh. Lecture Notes from TTM 4133 Long Term Evolution (LTE) 2016. Itslearning, [Online; Accesses 25.04.2017].

[Pul] Pulse Electronics. http://www.pulseelectronics.com/, [Online; Accesses 03.05.2017].

[Qua10] Qualcomm Incorporated. Enhancements, LTE Mobility, 2010.

[Rao11] Rao, V Srinivasa and Gajula, Rambabu. Protocol signaling procedures in LTE. *White Paper, Radisys Corporation*, 2011.

[Res] Ettus Research. USRP B200mini Data Sheet. Available: https://www.ettus.com/content/files/USRP_B200mini_Data_Sheet.pdf.

[Ret15] Torjus Bryne Retterstøl. Base Station Security Experiments Using USRP. Master's thesis, NTNU, 2015.

[RJP16] David Rupprecht, Kai Jansen, and Christina Pöpper. Putting LTE security functions to the test: a framework to evaluate implementation correctness. In *Proceedings of the 10th USENIX Conference on Offensive Technologies*, pages 40–51. USENIX Association, 2016.

[Rya] Ryan Gallagher. Meet the machines that steal your phone's data. https://arstechnica.com/tech-policy/2013/09/meet-the-machines-that-steal-your-phones-data/, [Online; Accesses 04.04.2017].

[SBA+15] Altaf Shaik, Ravishankar Borgaonkar, N Asokan, Valtteri Niemi, and Jean-Pierre Seifert. Practical attacks against privacy and availability in 4G/LTE mobile communication systems. *arXiv preprint arXiv:1510.07563*, 2015.

[SBT11] Stefania Sesia, Matthew Baker, and Issam Toufik. *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[Shaa] ShareTechnote. Multi Cell - Measurement in LTE. http://www.sharetechnote.com/html/Handbook_LTE_MultiCell_Measurement_LTE.html, [Online; Accesses 19.05.2017].

[Shab] ShareTechnote. Paging. http://www.sharetechnote.com/html/Paging_LTE.html, [Online; Accesses 10.05.2017].

[Sha16] Sharma, Purnima K and Sharma, Dinesh and Gupta, Akanksha. Cell Coverage Area and Link Budget Calculations in LTE System. *Indian Journal of Science and Technology*, 9(S1), 2016.

[SIM11] SIMalliance. UICC in LTE: A Guidance from SIMalliance. Ts, SIMalliance, February 2011.

[Sin12] Singh, Yuvraj. Comparison of Okumura, Hata and COST-231 Models on the Basis of Path Loss and Signal Strength. *International Journal of Computer Applications*, 59(11), 2012.

[Sno] SnoopSnitch Software. SnoopSnitch. https://opensource.srlabs.de/projects/snoopsnitch, [Online; Accesses 07.06.2017].

[Sof] Software Radio Systems. http://www.softwareradiosystems.com/, [Online; Accesses 08.05.2017].

[SPL] SPLAT! software. http://www.qsl.net/kd2bd/splat.html, [Online; Accesses 03.05.2017].

[Sri12] Kamakshi Sridhar. Introduction to Evolved Packet Core (EPC): EPC Elements, protocols and procedures. Ts, Alcatel Lucent, August 2012.

[srs] srsLTE. Open source 3GPP LTE library. https://github.com/srsLTE/srsLTE, [Online; Accesses 14.05.2017].

[Str07] Strobel, Daehyun. IMSI Catcher. *Chair for Communication Security, Ruhr-Universität Bochum*, 14, 2007.

[Tut17] Tutorialspoint.com. LTE Communication Channels. Available: https://www.tutorialspoint.com/lte/lte_communication_channels.htm, 2017.

[Uda] Udar Swapnil. Darshakframework. https://github.com/darshakframework/darshak, [Online; Accesses 14.05.2017].

[Zah12] Zahid Ghadialy. MIM: Machine Identity Module - M2M SIM. http://blog.3g4g.co.uk/2012/05/mim-machine-identity-module-m2m-sim.html, [Online; Accesses 26.04.2017], may 2012.

Appendix A

OpenAirInterface Installation Guide

This appendix provides a description on how to connect a USRP-based eNodeB and an OpenAirInterface EPC with a commercial UE. The following installation guide is based on the tutorial provided by OpenAirInterface [Ope17]; however, own opinions and experiences during the installation process are described.

A.1 Operating System Prerequisites

OpenAirInterface is very sensitive to version numbers, Linux kernel, etc., is it therefore important to follow this guide carefully. Notably, it is recommended to use the development branch in the OpenAirInterface Github repositories [Ope17].

The Debian-based Linux operating system Ubuntu 14.04 is required for running OpenAirInterface. Furthermore, low-latency Linux kernel version 3.19 is required. Ubuntu 14.04 is open source and can be freely downloaded from Ubuntu's website [Can], kernel version 3.19 is installed using the following command:

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

These changes require that you restart your computer. Use the following command in terminal to make sure the low-latency kernel is successfully installed:

```
$ uname -a
```

Subsequently, power management in the Basic Input-Output System (BIOS) and Central Processing Unit (CPU) frequency scaling is removed. Add the line `GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_pstate=disable"` in /etc/default/-

grub, then type the following in the terminal:

```
$ sudo update-grub
```

Frequency scaling is removed using the following command:

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

To run the CPU in performance mode, add the line `GOVERNOR="performance"` to /etc/default/cpufrequtils. Subsequently, prevent the settings to be overwritten by executing the following command:

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

A.2 Install and Configure eNodeB and EPC

The first step is to configure the /etc/hostname. Assume that the hostname of the PC is "wirelessLab".

```
$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       wirelessLab.openair4G.eur    wirelessLab
127.0.33.1       hss.openair4G.eur      hss
```

The second step is to download the project and run the automated build scripts. *Git* is installed using the command `sudo apt-get install git`. Consequently, the OpenAirInterface project is downloaded as follows:

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

Subsequently, the script in `openairinterface/cmake_targets/build_oai` is used to build the eNodeB. Notably, the following command is used for a USRP-based eNodeB; however, other setups exists [Ope17].

```
$ ./build_oai -I --eNB -x --install-system-files -w USRP
```

The scripts in `openair-cn/SCRIPTS` are used to build the MME, the HSS, and the SPGW:

```
$ ./build_mme -i
$ ./build_hss -i
$ ./build_spgw -i
```

Subsequently, the configuration files for MME, HSS, and SPGW in `~/openair-cn/ETC` are copied to the executable directory `/usr/local/etc/oai`. Moreover, the most important configuration parameters of the eNodeB are highlighted below (`~/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tm1.usrpb210.conf`):

```
tracking_area_code  =  "1";
mobile_country_code =  "208";
mobile_network_code =  "93";


////////// 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;
    };
```

MME configuration parameters (`/usr/local/etc/oai/mme.conf`):

```
REALM = "openair4G.eur";

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

GUMMEI_LIST = (
        {MCC="208" ; MNC="93"; MME_GID="4" ; MME_CODE="1"; }
     );

TAI_LIST = (
        {MCC="208" ; MNC="93";  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 :
{
    SGW_IPV4_ADDRESS_FOR_S11              = "127.0.8.1/8";
};
```

MME freediameter configuration parameters (`~/oai/freeDiameter/mme_fd.conf`):

```
Identity = "wirelessLab.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";
};
```

SPGW configuration parameters (`/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;

        SGW_INTERFACE_NAME_FOR_S5_S8_UP     = "none";
        SGW_IPV4_ADDRESS_FOR_S5_S8_UP       = "0.0.0.0/24";
    }
}
P-GW =
{
    NETWORK_INTERFACES :
    {
        PGW_INTERFACE_NAME_FOR_S5_S8 = "none";
        PGW_IPV4_ADDRESS_FOR_S5_S8   = "0.0.0.0/24";
        PGW_INTERFACE_NAME_FOR_SGI   = "eth0";
        PGW_IPV4_ADDRESS_FOR_SGI     = "129.241.208.234/23";
        PGW_MASQUERADE_SGI           = "yes";
    };
}
```

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

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

HSS freediameter configuration parameters (`~/oai/freeDiameter/hss_fd.conf`):

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

A.3 Run eNodeB and EPC

The eNodeB and the EPC are dependant of valid certificates in order to run. The required certificates are installed using the following command:

```
$ 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/ wirelessLab.openair4G.eur
```

Finally, compile and execute the network entities. Notably, always run the HSS first:

```
$ cd ~/openair-cn/SCRIPTS
$ ./build_hss -c
$ ./run_hss -i ~/openair-cn/SRC/OAI_HSS/db/oai_db.sql
```

Compile and run the MME:

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

Compile and run the SPGW:

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

Compile and run the eNodeB:

```
$ 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
  ~/targets/PROJECTS/GENERIC-LTE-EPC/CONF
  /enb.band7.tm1.usrpb210.conf -d
```

A.4 Configure OpenAirInterface as UE

The following steps are applied to configure OpenAirInterface as a UE. The prerequisites listed in Section A.1 applies.

```
$ cd ~/openairinterface5g/
$ source oaienv
$ cd cmake_targets
$ ./build_oai -w USRP --eNB --noS1 -x
```

A.5 Troubleshooting

The following steps were used for troubleshooting:

1. Check that the current OS version is Ubuntu 14.04 LTS and kernel version 3.19.0-61-lowlatency is installed
2. Make sure the fully qualified domain name is correctly configured in `/etc/host`
3. Confirm that the certificate from `gitlab.eurecom.fr` is correctly added to your Ubuntu installation
4. Mobile Network Code (MNC), Mobile Country Code (MCC) and Tracking Area Code (TAC) should be similar in `enb.band7.tm1.usrpb210.conf` and `mme.conf`

5. HSS hostname is set to *hss* in `mme.conf`
6. Make sure the PGW_IPV4_ADDRESS_FOR_SGI variable is set to the local IP address of your PC, in the `spgw.conf` file
7. Identity and realm should be properly configured in the `hss_fd.conf` file
8. Identity, realm, and connectionPeer should be properly configured in the `mme_fd.conf` file
9. MySQL username and password should be the same as provided during the installation, it is recommended to use the password *linux* as this is the default password used by OpenAirInterface
10. Make sure MME and HSS certificates are installed
11. If none of the steps above solves the problem try to compare your configuration files with example-configuration files provided by OpenAirInterface [Opea]

Appendix B
LTE IMSI Catcher Configuration Guide

This appendix contains all the necessary code-changes for configuring OpenAirInterface as an LTE IMSI Catcher.

For simplicity, all incoming *TAU Requests* are rejected using *TAU Reject* message with EMM rejection cause 10 (implicitly detached). Notably, OpenAirInterface has not implemented the complete TAU procedure and always returns a *TAU Reject* message [Bhe16]. The EMM rejection cause is defined in the file `openair-cn/SRC/NAS/EMM/nas_proc.c`:

```
if ( ue_ctx) {
    rc = emm_proc_tracking_area_update_reject
    (ue_id, EMM_CAUSE_IMPLICITLY_DETACHED);

    OAILOG_FUNC_RETURN (LOG_NAS_EMM, rc);
} else {
    rc = emm_proc_tracking_area_update_reject
    (ue_id, EMM_CAUSE_IMPLICITLY_DETACHED);

    OAILOG_FUNC_RETURN (LOG_NAS_EMM, rc);
}
```

The HSS does not contain subscriber data and hence no valid mapping between GUTI and IMSI. Consequently, no changes to the code are needed for triggering the *Identity Request* message. However, the file `openair-cn/SRC/NAS/EMM/SAP/emm_as.c` contains the original code written by OpenAirInterface:

```
if (emm_msg )
  switch (msg->msg_type) {
  case EMM_AS_MSG_TYPE_IDENT:
    if (msg->guti) {
        MSC_LOG_EVENT (MSC_NAS_EMM_MME,
        "send IDENTITY_REQUEST to s_TMSI %u.%u ",
        as_msg->s_tmsi.mme_code,
        as_msg->s_tmsi.m_tmsi);
    } else {
        MSC_LOG_EVENT (MSC_NAS_EMM_MME,
        "send IDENTITY_REQUEST to ue id" MME_UE_S1AP_ID_FMT,
        as_msg->ue_id);
    }

    size = emm_send_identity_request
    (msg, &emm_msg->identity_request);
    break;
```

As the HSS does not contain subscriber data for any UEs, it will always return DIAMETER_AUTHENTICATION_DATA_UNAVAILABLE. Consequently, the MME always returns the *Attach Reject* message with EMM rejection cause 15 (No suitable cells in TA). The MME rejection cause is defined in the file `openair-cn/SRC/NAS/nas_proc.c`:

```
switch (s6a_error) {
case DIAMETER_AUTHENTICATION_DATA_UNAVAILABLE:
  return NAS_CAUSE_NO_SUITABLE_CELLS_IN_TRACKING_AREA;
```

Appendix C
EMM Rejection Causes

This appendix provides a list of all the EMM rejection causes defined by the 3GPP standardization [3GP11c].

Table C.1: EMM rejection causes [3GP11c].

Cause Value	Cause Description
#2	IMSI unknown in HSS
#3	Illegal UE
#6	Illegal ME
#7	EPS services not allowed
#8	EPS services and non-EPS services not allowed
#9	UE identity cannot be derived by the network
#10	Implicitly detached
#11	PLMN not allowed
#12	Tracking Area not allowed
#13	Roaming not allowed in this tracking area
#14	EPS services not allowed in this PLMN
#15	No Suitable Cells In tracking area
#16	MSC temporarily not reachable
#17	Network failure
#18	CS domain not available
#19	ESM failure
#20	MAC failure
#21	Synch failure
#22	Congestion
#23	UE security capabilities mismatch
#24	Security mode rejected, unspecified
#25	Not authorized for this CSG
#26	Non-EPS authentication unacceptable
#38	CS fallback call establishment not allowed
#39	CS domain temporarily not available
#40	No EPS bearer context activated
#95	Semantically incorrect message
#96	Invalid mandatory information
#97	Message type non-existent or not implemented
#98	Message type not compatible with the protocol state
#99	Information element non-existent or not implemented
#100	Conditional IE error
#101	Message not compatible with the protocol state
#111	Protocol error, unspecified

Appendix D
Attach Procedure Time Calculation

This appendix provides a foundation for the calculations used to find the average time a UE is connected to an LTE IMSI Catcher. The calculations are based on the time interval between an *Attach Request* message and an *Attach Reject* message, measured from the IMSI Catcher side.

The goal of this experiment was to confirm that the duration of the attach procedure was small enough not to invoke suspicion by subscribers.

D.1 Attach Procedure Data

To be able to collect attach procedure data, the same experimental setup as described in Section 3.3 was used. Table D.1 summarizes the obtained results.

Table D.1: Collection of attach procedure data.

Test case	**Attach Request Received (CET)**	**Attach Reject Sent (CET)**	**Result (seconds)**
# 1	19:16:53.455007	19:16:53.477655	0.022648
# 2	19:17:00.176021	19:17:00.198921	0.022900
# 3	19:17:01.222881	19:17:01.240201	0.017320
# 4	19:32:41.187774	19:32:41.217387	0.029613
# 5	19:32:51.376803	19:32:51.402133	0.025330
Average attach procedure (seconds)			0.0235622

Appendix E
Decoding Paging Messages

This appendix includes the code used to decode paging messages. The decoding is a two-step procedure: the first step catches and decode paging message from PDSCH to raw ASN.1 hexadecimal format, the second step decodes the raw data to readable XML format.

E.1 PDSCH Decoding

The code snippet below catches and decodes paging messages in the PDSCH channel to raw ASN.1 data. The C code is originally written by SRS[1] [Sof]; however, the changes below have been made to induce desired behavior. The variable list in Table E.1 helps provide a better understanding of the code.

Table E.1: Variable list for the PDSCH decoder.

Name	Description
n	Integer indicating the data packet (if the value is greater that 1, a data packet is found)
data	Pointer containing the paging packet
srslte_vec_fprint_byte	Function used to print the hex string to stdout

[1] File srslte/examples/pdsch_ue.c

```
if (n < 0) {
// fprintf(stderr, "Error decoding UE DL\n");fflush(stdout);
} else if (n > 0) {
  //If the value of n is > 0 a "data" package is found
  srslte_vec_fprint_byte(stdout, data, n/8);;
  /* Send data if socket active */
  if (prog_args.net_port > 0) {
    srslte_netsink_write(&net_sink, data, 1+(n-1)/8);
  }

  #ifdef PRINT_CHANGE_SCHEDULIGN
  if (ue_dl.dl_dci.mcs_idx != old_dl_dci.mcs_idx ||
      memcmp(&ue_dl.dl_dci.type0_alloc,
      &old_dl_dci.type0_alloc, sizeof(srslte_ra_type0_t)) ||
      memcmp(&ue_dl.dl_dci.type1_alloc,
      &old_dl_dci.type1_alloc, sizeof(srslte_ra_type1_t)) ||
      memcmp(&ue_dl.dl_dci.type2_alloc,
      &old_dl_dci.type2_alloc, sizeof(srslte_ra_type2_t)))
  {
    memcpy(&old_dl_dci, &ue_dl.dl_dci,
    sizeof(srslte_ra_dl_dci_t));
    fflush(stdout);
    printf("Format: \%s\n",
    srslte_dci_format_string(ue_dl.dci_format));
    srslte_ra_pdsch_fprint(stdout,
    &old_dl_dci, cell.nof_prb);
    srslte_ra_dl_grant_fprint(stdout,
    &ue_dl.pdsch_cfg.grant);
  }
  #endif

}
```

E.2 ASN.1 Decoding

The python script below decodes raw ASN.1 data to XML format[2]. The script uses the external library *libmich*, which handles the actual decoding part [Lib]. The variable list given in Table E.2 helps provide a better understanding of the code.

[2]Python was chosen for the ASN.1 decoding because it contains dynamic external libraries.

Table E.2: Variable list for the ASN.1 decoder.

Name	Description
pcch	Defines the ASN.1 interface
buf	Stores the hex string temporarily
line	Stores the decoded value

```
#!/usr/bin/python

from libmich.asn1.processor import *;

def decodePCCH(pcchHex):
    load_module('RRCLTE');
    ASN1.ASN1Obj.CODEC = PER;
    PER.VARIANT = 'U';
    pcch = GLOBAL.TYPE['PCCH-Message'];
    buf = pcchHex.decode('hex');
    pcch.decode(buf);
    show(pcch);

with open('/srsLTE/1hour_ice_796MHz/UE.log') as fp:
    for line in fp:
        if line.startswith("["):
            line = line[1:-4] #Stripping line
            line = line.replace(" ", "")
            decodePCCH(line);
```

Appendix E
Results Gathered from SIB Type 1-7

This Appendix provides the full content of the SIB 1-7 messages broadcasted by Telia's cell 34767628. SIB messages from Telenor and ice.net cells are not included in this appendix as the structure is the same as for the Telia cell; however, Section 4.4.5 summarizes the most important parameters for all PLMNs. SIB message type 1-7 are structured using XML notation.

SIB type 1

```
<BCCH-DL-SCH-Message>
    <message>
        <c1>
            <systemInformationBlockType1>
                <cellAccessRelatedInfo>
                    <plmn-IdentityList>
                        <PLMN-IdentityInfo>
                            <plmn-Identity>
                                <mcc>
                                    <MCC-MNC-Digit>
                                        2
                                    </MCC-MNC-Digit>
                                    <MCC-MNC-Digit>
                                        4
                                    </MCC-MNC-Digit>
                                    <MCC-MNC-Digit>
                                        2
                                    </MCC-MNC-Digit>
                                </mcc>
                                <mnc>
                                    <MCC-MNC-Digit>
```

```
                                    0
                                </MCC-MNC-Digit>
                                <MCC-MNC-Digit>
                                    2
                                </MCC-MNC-Digit>
                            </mnc>
                        </plmn-Identity>
                        <cellReservedForOperatorUse>
                            <notReserved/>
                        </cellReservedForOperatorUse>
                    </PLMN-IdentityInfo>
                </plmn-IdentityList>
                <trackingAreaCode>
                    0000100100000001
                </trackingAreaCode>
                <cellIdentity>
                    00100001001010000011000011 00
                </cellIdentity>
                <cellBarred><notBarred/></cellBarred>
                <intraFreqReselection>
                    <allowed/>
                </intraFreqReselection>
                <csg-Indication>
                    <false/>
                </csg-Indication>
            </cellAccessRelatedInfo>
            <cellSelectionInfo>
                <q-RxLevMin>-66</q-RxLevMin>
            </cellSelectionInfo>
            <freqBandIndicator>20</freqBandIndicator>
            <schedulingInfoList>
                <SchedulingInfo>
                    <si-Periodicity>
                        <rf8/>
                    </si-Periodicity>
                    <sib-MappingInfo><sibType3/>
                    </sib-MappingInfo>
                </SchedulingInfo>
                <SchedulingInfo>
                    <si-Periodicity>
                        <rf64/>
```

```
                    </si-Periodicity>
                    <sib-MappingInfo><sibType4/>
                    </sib-MappingInfo>
                </SchedulingInfo>
                <SchedulingInfo>
                    <si-Periodicity>
                        <rf64/>
                    </si-Periodicity>
                    <sib-MappingInfo><sibType5/>
                    </sib-MappingInfo>
                </SchedulingInfo>
                <SchedulingInfo>
                    <si-Periodicity>
                        <rf64/>
                    </si-Periodicity>
                    <sib-MappingInfo><sibType6/>
                    </sib-MappingInfo>
                </SchedulingInfo>
                <SchedulingInfo>
                    <si-Periodicity>
                        <rf64/>
                    </si-Periodicity>
                    <sib-MappingInfo><sibType7/>
                    </sib-MappingInfo>
                </SchedulingInfo>
            </schedulingInfoList>
            <si-WindowLength><ms10/></si-WindowLength>
            <systemInfoValueTag>0</systemInfoValueTag>
        </systemInformationBlockType1>
    </c1>
</message>
</BCCH-DL-SCH-Message>
```

SIB type 2

```
<sib2>
    <radioResourceConfigCommon>
        <rach-ConfigCommon>
            <preambleInfo>
                <numberOfRA-Preambles>
                    <n52/>
                </numberOfRA-Preambles>
            </preambleInfo>
            <powerRampingParameters>
                <powerRampingStep><dB4/></powerRampingStep>
                <preambleInitialReceivedTargetPower>
                    <dBm-110/>
                </preambleInitialReceivedTargetPower>
            </powerRampingParameters>
            <ra-SupervisionInfo>
                <preambleTransMax><n10/></preambleTransMax>
                <ra-ResponseWindowSize>
                    <sf10/>
                </ra-ResponseWindowSize>
                <mac-ContentionResolutionTimer>
                    <sf64/>
                </mac-ContentionResolutionTimer>
            </ra-SupervisionInfo>
            <maxHARQ-Msg3Tx>4</maxHARQ-Msg3Tx>
        </rach-ConfigCommon>
        <bcch-Config>
            <modificationPeriodCoeff>
                <n2/>
            </modificationPeriodCoeff>
        </bcch-Config>
        <pcch-Config>
            <defaultPagingCycle>
                <rf128/>
            </defaultPagingCycle>
            <nB><oneT/></nB>
        </pcch-Config>
        <prach-Config>
            <rootSequenceIndex>576</rootSequenceIndex>
            <prach-ConfigInfo>
                <prach-ConfigIndex>19</prach-ConfigIndex>
```

```
            <highSpeedFlag><false/></highSpeedFlag>
            <zeroCorrelationZoneConfig>
                15
            </zeroCorrelationZoneConfig>
            <prach-FreqOffset>2</prach-FreqOffset>
        </prach-ConfigInfo>
    </prach-Config>
    <pdsch-ConfigCommon>
        <referenceSignalPower>18</referenceSignalPower>
        <p-b>1</p-b>
    </pdsch-ConfigCommon>
    <pusch-ConfigCommon>
        <pusch-ConfigBasic>
            <n-SB>1</n-SB>
            <hoppingMode><interSubFrame/></hoppingMode>
            <pusch-HoppingOffset>0</pusch-HoppingOffset>
            <enable64QAM><true/></enable64QAM>
        </pusch-ConfigBasic>
        <ul-ReferenceSignalsPUSCH>
            <groupHoppingEnabled>
                <true/>
            </groupHoppingEnabled>
            <groupAssignmentPUSCH>
                0
            </groupAssignmentPUSCH>
            <sequenceHoppingEnabled>
                <false/>
            </sequenceHoppingEnabled>
            <cyclicShift>0</cyclicShift>
        </ul-ReferenceSignalsPUSCH>
    </pusch-ConfigCommon>
    <pucch-ConfigCommon>
        <deltaPUCCH-Shift><ds1/></deltaPUCCH-Shift>
        <nRB-CQI>2</nRB-CQI>
        <nCS-AN>0</nCS-AN>
        <n1PUCCH-AN>8</n1PUCCH-AN>
    </pucch-ConfigCommon>
    <soundingRS-UL-ConfigCommon>
        <release></release>
    </soundingRS-UL-ConfigCommon>
    <uplinkPowerControlCommon>
```

```
            <p0-NominalPUSCH>-97</p0-NominalPUSCH>
            <alpha><al1/></alpha>
            <p0-NominalPUCCH>-110</p0-NominalPUCCH>
            <deltaFList-PUCCH>
                <deltaF-PUCCH-Format1>
                    <deltaF0/>
                </deltaF-PUCCH-Format1>
                <deltaF-PUCCH-Format1b>
                    <deltaF3/>
                </deltaF-PUCCH-Format1b>
                <deltaF-PUCCH-Format2>
                    <deltaF0/>
                </deltaF-PUCCH-Format2>
                <deltaF-PUCCH-Format2a>
                    <deltaF0/>
                </deltaF-PUCCH-Format2a>
                <deltaF-PUCCH-Format2b>
                    <deltaF0/>
                </deltaF-PUCCH-Format2b>
            </deltaFList-PUCCH>
            <deltaPreambleMsg3>6</deltaPreambleMsg3>
        </uplinkPowerControlCommon>
        <ul-CyclicPrefixLength>
            <len1/>
        </ul-CyclicPrefixLength>
        <ext1>
            <uplinkPowerControlCommon-v1020>
                <deltaF-PUCCH-Format3-r10>
                    <deltaF0/>
                </deltaF-PUCCH-Format3-r10>
                <deltaF-PUCCH-Format1bCS-r10>
                    <deltaF2/>
                </deltaF-PUCCH-Format1bCS-r10>
            </uplinkPowerControlCommon-v1020>
        </ext1>
    </radioResourceConfigCommon>
    <ue-TimersAndConstants>
        <t300><ms1000/></t300>
        <t301><ms400/></t301>
        <t310><ms2000/></t310>
        <n310><n20/></n310>
```

```
        <t311><ms3000/></t311>
        <n311><n1/></n311>
    </ue-TimersAndConstants>
    <freqInfo>
        <additionalSpectrumEmission>
            1
        </additionalSpectrumEmission>
    </freqInfo>
    <timeAlignmentTimerCommon>
        <infinity/>
    </timeAlignmentTimerCommon>
</sib2>
```

SIB type 3

```
<sib3>
    <cellReselectionInfoCommon>
        <q-Hyst><dB4/></q-Hyst>
    </cellReselectionInfoCommon>
    <cellReselectionServingFreqInfo>
        <s-NonIntraSearch>8</s-NonIntraSearch>
        <threshServingLow>5</threshServingLow>
        <cellReselectionPriority>5</cellReselectionPriority>
    </cellReselectionServingFreqInfo>
    <intraFreqCellReselectionInfo>
        <q-RxLevMin>-66</q-RxLevMin>
        <s-IntraSearch>31</s-IntraSearch>
        <allowedMeasBandwidth>
            <mbw50/>
        </allowedMeasBandwidth>
        <presenceAntennaPort1>
            <true/>
        </presenceAntennaPort1>
        <neighCellConfig>
            10
        </neighCellConfig>
        <t-ReselectionEUTRA>1</t-ReselectionEUTRA>
    </intraFreqCellReselectionInfo>
</sib3>
```

SIB type 4

```
<BCCH-DL-SCH-Message>
<message>
<c1>
<systemInformation>
<criticalExtensions>
    <systemInformation-r8>
        <sib-TypeAndInfo>
            <sib4>
                <intraFreqNeighCellList>
                    <IntraFreqNeighCellInfo>
                        <physCellId>124</physCellId>
                        <q-OffsetCell><dB3/></q-OffsetCell>
                    </IntraFreqNeighCellInfo>
                    <IntraFreqNeighCellInfo>
                        <physCellId>200</physCellId>
                        <q-OffsetCell><dB2/></q-OffsetCell>
                    </IntraFreqNeighCellInfo>
                    <IntraFreqNeighCellInfo>
                        <physCellId>225</physCellId>
                        <q-OffsetCell><dB2/></q-OffsetCell>
                    </IntraFreqNeighCellInfo>
                    <IntraFreqNeighCellInfo>
                        <physCellId>226</physCellId>
                        <q-OffsetCell><dB-2/></q-OffsetCell>
                    </IntraFreqNeighCellInfo>
                    <IntraFreqNeighCellInfo>
                        <physCellId>232</physCellId>
                        <q-OffsetCell><dB2/></q-OffsetCell>
                    </IntraFreqNeighCellInfo>
                    <IntraFreqNeighCellInfo>
                        <physCellId>231</physCellId>
                        <q-OffsetCell><dB-1/></q-OffsetCell>
                    </IntraFreqNeighCellInfo>
                </intraFreqNeighCellList>
            </sib4>
        </sib-TypeAndInfo>
    </systemInformation-r8>
</criticalExtensions>
</systemInformation>
</c1>
```

```
</message>
</BCCH-DL-SCH-Message>
```

SIB type 5

```
<BCCH-DL-SCH-Message>
<message>
<c1>
<systemInformation>
<criticalExtensions>
<systemInformation-r8>
<sib-TypeAndInfo>
    <sib5>
        <interFreqCarrierFreqList>
            <InterFreqCarrierFreqInfo>
                <dl-CarrierFreq>2850</dl-CarrierFreq>
                <q-RxLevMin>-66</q-RxLevMin>
                <t-ReselectionEUTRA>1</t-ReselectionEUTRA>
                <threshX-High>8</threshX-High>
                <threshX-Low>8</threshX-Low>
                <allowedMeasBandwidth>
                    <mbw100/>
                </allowedMeasBandwidth>
                <presenceAntennaPort1>
                    <true/>
                </presenceAntennaPort1>
                <cellReselectionPriority>
                    7
                </cellReselectionPriority>
                <neighCellConfig>
                    10
                </neighCellConfig>
                <q-OffsetFreq><dB0/></q-OffsetFreq>
                <interFreqNeighCellList>
                    <InterFreqNeighCellInfo>
                        <physCellId>123</physCellId>
                        <q-OffsetCell><dB-1/></q-OffsetCell>
                    </InterFreqNeighCellInfo>
                </interFreqNeighCellList>
            </InterFreqCarrierFreqInfo>
            <InterFreqCarrierFreqInfo>
                <dl-CarrierFreq>1650</dl-CarrierFreq>
```

```
                    <q-RxLevMin>-66</q-RxLevMin>
                    <t-ReselectionEUTRA>1</t-ReselectionEUTRA>
                    <threshX-High>8</threshX-High>
                    <threshX-Low>8</threshX-Low>
                    <allowedMeasBandwidth>
                        <mbw100/>
                    </allowedMeasBandwidth>
                    <presenceAntennaPort1>
                        <true/>
                    </presenceAntennaPort1>
                    <cellReselectionPriority>
                        6
                    </cellReselectionPriority>
                    <neighCellConfig>
                        10
                    </neighCellConfig>
                    <q-OffsetFreq><dB0/></q-OffsetFreq>
                    <interFreqNeighCellList>
                        <InterFreqNeighCellInfo>
                            <physCellId>47</physCellId>
                            <q-OffsetCell><dB-2/></q-OffsetCell>
                        </InterFreqNeighCellInfo>
                        <InterFreqNeighCellInfo>
                            <physCellId>125</physCellId>
                            <q-OffsetCell><dB-2/></q-OffsetCell>
                        </InterFreqNeighCellInfo>
                        <InterFreqNeighCellInfo>
                            <physCellId>123</physCellId>
                            <q-OffsetCell><dB1/></q-OffsetCell>
                        </InterFreqNeighCellInfo>
                    </interFreqNeighCellList>
                </InterFreqCarrierFreqInfo>
            </interFreqCarrierFreqList>
        </sib5>
    </sib-TypeAndInfo>
    </systemInformation-r8>
    </criticalExtensions>
    </systemInformation>
    </c1>
    </message>
    </BCCH-DL-SCH-Message>
```

SIB type 6

```
<BCCH-DL-SCH-Message>
<message>
<c1>
<systemInformation>
<criticalExtensions>
    <systemInformation-r8>
        <sib-TypeAndInfo>
            <sib6>
                <carrierFreqListUTRA-FDD>
                    <CarrierFreqUTRA-FDD>
                        <carrierFreq>10588</carrierFreq>
                        <cellReselectionPriority>
                            3
                        </cellReselectionPriority>
                        <threshX-High>2</threshX-High>
                        <threshX-Low>5</threshX-Low>
                        <q-RxLevMin>-58</q-RxLevMin>
                        <p-MaxUTRA>24</p-MaxUTRA>
                        <q-QualMin>-18</q-QualMin>
                    </CarrierFreqUTRA-FDD>
                    <CarrierFreqUTRA-FDD>
                        <carrierFreq>2963</carrierFreq>
                        <cellReselectionPriority>
                            2
                        </cellReselectionPriority>
                        <threshX-High>2</threshX-High>
                        <threshX-Low>5</threshX-Low>
                        <q-RxLevMin>-58</q-RxLevMin>
                        <p-MaxUTRA>24</p-MaxUTRA>
                        <q-QualMin>-18</q-QualMin>
                    </CarrierFreqUTRA-FDD>
                </carrierFreqListUTRA-FDD>
                <t-ReselectionUTRA>
                    2
                </t-ReselectionUTRA>
            </sib6>
        </sib-TypeAndInfo>
    </systemInformation-r8>
</criticalExtensions>
</systemInformation>
```

```
</c1>
</message>
</BCCH-DL-SCH-Message>
```

SIB type 7

```
<BCCH-DL-SCH-Message>
<message>
<c1>
<systemInformation>
<criticalExtensions>
<systemInformation-r8>
<sib-TypeAndInfo>
<sib7>
    <t-ReselectionGERAN>2</t-ReselectionGERAN>
    <carrierFreqsInfoList>
    <CarrierFreqsInfoGERAN>
        <carrierFreqs>
            <startingARFCN>41</startingARFCN>
            <bandIndicator><dcs1800/></bandIndicator>
            <followingARFCNs>
                <explicitListOfARFCNs>
                    <ARFCN-ValueGERAN>21</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>29</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>47</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>24</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>5</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>4</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>38</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>8</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>10</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>34</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>11</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>12</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>14</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>26</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>42</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>7</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>25</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>36</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>17</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>13</ARFCN-ValueGERAN>
```

```
                    <ARFCN-ValueGERAN>45</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>28</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>48</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>32</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>3</ARFCN-ValueGERAN>
                </explicitListOfARFCNs>
            </followingARFCNs>
        </carrierFreqs>
        <commonInfo>
            <cellReselectionPriority>
                1
            </cellReselectionPriority>
            <ncc-Permitted>
                11111111
            </ncc-Permitted>
            <q-RxLevMin>3</q-RxLevMin>
            <p-MaxGERAN>33</p-MaxGERAN>
            <threshX-High>2</threshX-High>
            <threshX-Low>3</threshX-Low>
        </commonInfo>
    </CarrierFreqsInfoGERAN>
    <CarrierFreqsInfoGERAN>
        <carrierFreqs>
            <startingARFCN>1</startingARFCN>
            <bandIndicator><dcs1800/></bandIndicator>
            <followingARFCNs>
                <explicitListOfARFCNs>
                    <ARFCN-ValueGERAN>2</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>6</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>9</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>15</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>16</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>18</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>19</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>20</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>22</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>23</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>27</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>30</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>31</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>33</ARFCN-ValueGERAN>
```

```
                    <ARFCN-ValueGERAN>35</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>37</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>39</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>40</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>43</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>44</ARFCN-ValueGERAN>
                    <ARFCN-ValueGERAN>46</ARFCN-ValueGERAN>
                </explicitListOfARFCNs>
            </followingARFCNs>
        </carrierFreqs>
        <commonInfo>
            <cellReselectionPriority>
                1
            </cellReselectionPriority>
            <ncc-Permitted>
                11111111
            </ncc-Permitted>
            <q-RxLevMin>3</q-RxLevMin>
            <p-MaxGERAN>33</p-MaxGERAN>
            <threshX-High>2</threshX-High>
            <threshX-Low>3</threshX-Low>
        </commonInfo>
    </CarrierFreqsInfoGERAN>
    </carrierFreqsInfoList>
</sib7>
</sib-TypeAndInfo>
</systemInformation-r8>
</criticalExtensions>
</systemInformation>
</c1>
</message>
</BCCH-DL-SCH-Message>
```