



University of Piraeus
School of Information and Communication Technologies
Department of Digital Systems
Postgraduate Program of Studies
MSc Digital Systems Security

Master Thesis – Privacy Evaluation of 5G Networks

Supervisor Professor: Christos Xenakis

Mentor: Anna Angelogianni

Name-Surname	E-mail	Student ID.
Santorinaios Dimitris	d.santorinaios@ssl-unipi.gr	MTE2028

Piraeus

07/2022

Περίληψη

Η τεχνολογία 5G θεωρείται η λύση στην ογκώδη ζήτηση υπηρεσιών από απλούς χρήστες μέχρι και την βιομηχανία. Με ταχύτητες που ξεπερνούν με διαφορά τον προκάτοχο του και πολύ μικρότερο χρόνο καθυστέρηση μεταφοράς, δίνει την δυνατότητα στους χρήστες, IoT συσκευές, αυτοκίνητα, βιομηχανικό εξοπλισμό, να έχουν πρόσβαση στο internet και να μεταφέρουν πληροφορίες σε πραγματικό χρόνο. Η πρόκληση για διασφάλιση της πληροφορίας και η ιδιωτικότητα των χρηστών, έχει ως συνέπεια να γίνει μια σημαντική προσπάθεια για την βελτίωση τους. Επομένως ένα από τα αντικείμενα αυτής της διπλωματικής είναι να παρουσιάσει την αρχιτεκτονική του 5G. Επιπλέον, χρησιμοποιώντας ένα εργαλείο που βοηθάει στην παρακολούθηση και την ανάλυση ασύρματων μηνυμάτων, θα μελετηθεί η παρούσα κατάσταση της υλοποίησης του 5G στην Ελλάδα για μερικούς παρόχους τηλεπικοινωνιών.

Abstract

5G technology is considered the solution to the vast demand for services not only from ordinary users, but also the industry. With speeds and latency far exceeding those provided by its predecessor, it enables internet access to users and IoT devices, cars, or industrial machines, communicating in real time. As securing the information and the user privacy has become one of the biggest challenges, a considerable effort has been made for its improvement. Therefore, one of the objectives of this thesis is to display the overall architecture of the 5G. Furthermore, by using a tool that helps the interception and the analysis of radio messages, an investigation will be conducted for better understanding the current phase of the 5G implementation in Greece for some mobile network operators.

Table of Content

Table of Figures	1
Table of Tables	2
Abbreviations	3
Introduction	6
Theoretical background	7
Cellular networks & 5G	7
3GPP & 5G	7
Architecture	8
NSA	10
4G architecture	11
SA	12
5G NR Radio Protocol Stack	15
5G SA Call Flow	18
Authentication Key Agreement in 4G and 5G	22
4G Authentication	23
4G EPS-AKA	23
5G Authentication	24
5G Authentication Framework	25
5G-AKA	26
EAP-AKA'	28
EAP-TLS	28
Subscriber Privacy	30
Identifiers in 5G	31
Evaluation of MNOs 5G deployment	34
Mobile Network Inspection Tools	34
MobileInsight	35
Solution Approach	35
Application Architecture	36
Mobile Interface	37

Desktop Interface	40
Evaluation of Captured Messages	46
Results	47
Summary	52
Conclusion	53
Scripts	54
Custom Plugin for MobileInsight Desktop Interface	54
Custom Plugin for MobileInsight Mobile Interface	56
References	58

Table of Figures

Figure 1- Evolution of telecommunications	9
Figure 2 - Cosmote 5G NSA mode (left) & Vodafone 5G NSA mode (right)	10
Figure 3 - 5G NSA architecture	11
Figure 4- 4G architecture	12
Figure 5- 5G architecture	13
Figure 6 - NR Protocol Stack User Plane	15
Figure 7 - NR Protocol Stack Control Plane.....	16
Figure 8- 5G SA register call procedure	19
Figure 9- 4G Authentication Procedure	24
Figure 10- 5G Authentication Framework.....	26
Figure 11- 5G Authentication Procedure	27
Figure 12- Key Hierarchy in 4G and 5G	29
Figure 13 - SUPI structure	31
Figure 14- SUCI structure.....	32
Figure 15-GUTI structure	33
Figure 16- MobileInsight high level architecture	36
Figure 17- MobileInsight App Home (left) & Menu (right).....	37
Figure 18- MobileInsight plugins (left) & LogType (right)	38
Figure 19- MobileInsight custom plugin	39
Figure 20- ADB Diagnostic Mode.....	41
Figure 21- lsusb output before diagnostic mode.....	41
Figure 22- lsusb after diagnostic mode	42
Figure 23- list ttyUSBx devices	42
Figure 24- MobileInsight monitor before capture	43
Figure 25- MobileInsight captured messages	44
Figure 26- MobileInsight offline analysis.....	44
Figure 27- output file of offline-analysis.py	45
Figure 28- MobileInsight GUI (mi-gui).....	45
Figure 29- Amarisoft GUI 5G-TMSI.....	46
Figure 30- mi-gui search RAND.....	48

Figure 31- Vodafone IMSI sent 26 times51

Table of Tables

Table 1- Cipherring and integrity algorithms used from Vodafone & Cosmote47
Table 2 - AKA frequency49
Table 3 - TMSIs captured from Cosmote50
Table 4- TMSIs captured from Vodafone.....50
Table 5- TMSIs captured from Amarisoft50

Abbreviations

5G	Fifth generation
IMT	International Mobile Telecommunications
LTE	Long Term Evolution
3GPP	Third Generation Partnership Project
GSMA	Groupe Speciale Mobile Association
5G NR	Fifth generation New Radio
MNO	Mobile Network Operator
NSA	Non-Stand Alone
AN	Access Network
EPC	Evolved Packet Core
SA	Stand-Alone
CN	Core Network
CUPS	Control and User plane separation
SDN	Software Defined Networking
CNA	Cloud Native Architecture
BS	Base Station
RAN	Radio Access Network
gNB	Next Generation Node B
eNB	Evolved Node B
UE	User Equipment
MME	Mobility Management Entity
SGW	Serving Gateway
E-UTRAN	Evolved-UMTS Terrestrial Radio Access Network
UMTS	Universal Mobile Telecommunications Service
PGW	Packet Data Network Gateway
HSS	Home Subscriber Server
USIM	Universal Subscriber Identity Module
IMSI	International Mobile Subscriber Identity
ICCID	Integrated Circuit Card Identifier
PCRF	Policy Control and Charging Rules Function
PCEF	Policy Control Enforcement Function
SUPI	Subscription Permanent Identifier
SBA	Service-Based Architecture
SDAP	Service Data Adaptation Protocol
AMF	Access and Mobility Management Function
AUSF	Authentication Server Function
NEF	Network Exposure Function
UDM	Unified Data Management
NRF	Network Repository Function
PCF	Policy Control Function
SMF	Session Management Function

AF	Application Function
UPF	User Plane Function
DN	Data Network
AS	Access Stratum
RRC	Radio Resource Control
SRB	Signaling Radio Bearers
DRB	Data Radio Bearers
QoS	Quality of Service
NAS	Non-Access Stratum
NF	Network Functions
AS	Access Stratum
EAP	Extensible Authentication Protocol
DHCP	Dynamic Host Configuration Protocol
RAT	Radio Access Technology
CP	Control Plane
UDR	Unified Data Repository
PDU	Protocol Data Unit
SDU	Service Data Unit
RLC-AM	Radio Link Control Acknowledge Mode
ARQ	Automatic Repeat Request
HARQ	Hybrid Automatic Repeat Request
TB	Transport Blocks
PEI	Permanent Equipment Identifier
SUCI	Subscriber Concealed Identifier
OTA	Over The Air
NGAP	New Generation Application Protocol
SSB	Synchronization Signal Block
RACH	Random Access Procedure
MIB	Master Information Block
PBCH	Physical Broadcast Channel
OFDM	Orthogonal Frequency Division Multiplexing
RAPID	Random Access Preamble Identifier
DCI	Downlink Control Information
CRC	Cyclic redundancy check
RNTI	Radio Network Temporary Identifier
MCS	Modulation Coding Scheme
RAR	Random Access Response
PDSCH	Physical downlink control channel
UL_CCCH	Uplink Common Control Channel
PLMN-ID	Public Land Mobile Network Identity
S-NSSAI	Single Network Slice Selection Assistance Information
IMEI	International Mobile Equipment Identity
IMEISV	IMEI software version
MCC	Mobile Country Code
MNC	Mobile Network Code
MSIN	Mobile Subscriber Identification Number
NAI	Network Access Identifier
EPS-AKA	Evolved Packet System based Authentication and Key Agreement

UICC	Universal Integrated Circuit Card
AV	Authentication Vectors
SEAF	Security Anchor Function
N3IWF	Non-3GPP Interworking Function
SIDF	Subscription Identifier De-concealing Function
ARPF	Authentication Credential Repository and Processing Function
TLS	Transport Layer Security
PSK	Pre-Shared Key
EMSK	Extended Master Session Key
CK	Cipher Key
K_i	Shared Secret Key
IK	Integrity Key
GUTI	Global Unique Temporary Identifier
TMSI	Temporary Mobile Subscriber Identity
GUAMI	Globally Unique AMF Identifier
API	Application Programming Interface
OS	Operating System
ADB	Anrdoid Debug Bridge

Introduction

The fifth generation of mobile telecommunications (5G) is expected to be one the most important innovations of the current decade. 5G is the motor that will open a new era of accessibility, quality, and reliability to everyone. 5G is expected to deliver technological advances with low latency, high speed, and reliable connections to mobile autonomous systems and large-scale deployments of IoT devices for Machine-Type Communications. 5G is the first mobile architecture designed to support multiple, specific use cases, each with its own unique cybersecurity requirements. For example, 5G will enable massive IoT applications, such as traffic sensors and vehicle-to-infrastructure services, that are the foundation for smart cities.

A critical issue that has arisen is that except for the vulnerabilities that are constantly being discovered, there are inconsistent implementations of 5G security in deployed and planned 5G networks from Mobile Network Operators (MNOs). Although 3GPP security specifications features and functions are required to be supported by vendors, they are all optional for 5G service providers, because some countries do not want these security features implemented by their national telecommunications as these security features also provide privacy.

Theoretical background

Cellular networks & 5G

A Cellular network or Mobile network is a radio network distributed over land areas called cells, each served by at least one fixed-location transceiver, known as a cell site or base station. In a cellular network, each cell uses a different set of frequencies from neighboring cells, to avoid interference and provide guaranteed bandwidth within each cell. Like its predecessors, 5G networks are cellular networks where the devices in a cell are connected to the Internet and telephone network by radio waves through a local antenna in the cell. Several organizations have defined specifications for each generation of cellular networks trying to establish standards for each one. According to International Mobile Telecommunications 2020 Standard (IMT-2020), within the 5G standards is a minimum speed that must be achieved for an operator to be classified as having a “5G network” and this has been set at 20Gbps downlink and 10Gbps uplink per mobile base station [1]. These are the minimum download and upload speeds that must be achievable by a single cell on a network. They are not the minimum speeds consumers will see. Of course, in the real world where billions of devices are connected, receiving and sending signals simultaneously, these maximum speeds are out of reach. As a result, the 5G specifications call for a per-user download speed of 100Mbps and 50Mbps upload. This is close to what is achievable on Long Term Evolution (LTE) networks, although while these speeds are rare on LTE, they are likely to be standard on 5G.

3GPP & 5G

For each iteration of mobile technology, standards are put in place by a group called Third Generation Partnership Project (3GPP) to make sure the technology reaches a certain set of benchmarks, to protect consumers and regulate how the technology is deployed and used. 3GPP is the standards body for mobile telecoms and, as its name suggests, the partnership was originally put together for 3G in 1998 [2]. It combines the standards bodies from key areas - such as Europe - and territories - such as South Korea,

Japan and China. Other key partners are organizations that bind together other organizations with an interest in telecommunications standards. These include the Groupe Speciale Mobile Association (GSMA) - the trade body that joins together mobile operators - and the Wireless Broadband Alliance, intended to further interoperability between Wi-Fi standards.

The major focus of 3GPP is to make the system backwards and forwards compatible where possible, to ensure that the operation of user equipment is uninterrupted as generations and releases advance. For 5G, many operators are starting with dual connectivity between LTE and 5G New Radio (NR) equipment - using the 'non-Standalone' architecture. In the process of completing the early drop of 5G NR care has been taken to build 'forward compatibility' into Non-Standalone NR equipment, to ensure that it will be fit for use on Standalone 5G NR systems.

Architecture

As radio generations advance (Figure 1), with each generation lasting about a decade, MNOs are faced with costly transformations, both in experienced personnel and equipment. The road to 5G, as it turns out, is not a straight line, and consideration for backward compatibility must be anticipated. 3GPP defines two deployment options for 5G:

1. The “Non-Stand Alone” (NSA) architecture, where the 5G Radio Access Network (AN) and its NR interface is used in conjunction with the existing LTE and evolved packet core (EPC) infrastructure Core Network (CN), thus making the NR technology available without network replacement. In this configuration, only the 4G services are supported, but enjoying the capacities offered by the 5G New Radio.

2. The “Stand-Alone” (SA) architecture, where the NR is connected to the 5G CN. Only in this configuration, the full set of 5G services are supported.

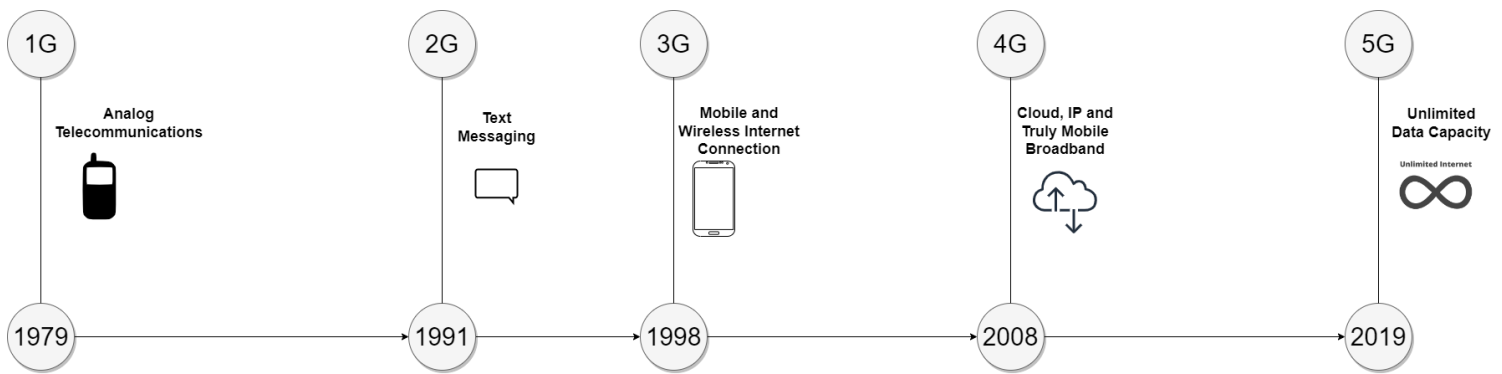


Figure 1- Evolution of telecommunications

MNOs plan for 5G deployment will be mostly stipulated on the finance factor. As virtualization was not mandated during the deployment of 4G, operators approached the deployment from a broad spectrum utilizing proprietary and virtualized solutions. With 5G, virtualization is essential. MNOs now have an opportunity to transform the way they build and operate their networks.

For operators looking to deliver high-speed connectivity to consumers with 5G-enabled devices, an NSA architecture is more appealing, because it enables them to leverage their existing network investments in transport and mobile core rather than deploy a completely new end to end 5G network. This can be combined with efforts to reduce network operating costs by adopting virtualization and control and user plane separation (CUPS) using software-defined networking (SDN).

However, for some MNOs who have their sights set on new enterprise 5G services such as smart cities, smart factories, or other vertical market solutions, a SA architecture could make more sense. As 3GPP has now standardized, MNOs can build an entirely new fully virtualized 5G network that includes a new radio access network, new transport network, and new 5G mobile core and edge networks – standing alone and separate from their existing 4G and legacy networks. 5G SA architecture is a fully virtualized, cloud-native architecture (CNA) that introduces new ways to develop, deploy, and manage services. CNA includes concepts of microservices and service-based interfaces that greatly simplify services, dramatically reducing the cost of operation and speeding up the introduction of new revenue-generating services.

It is worth noting that NSA and SA are not an either/or proposition, but more of a “sooner or later” consideration. MNOs that begin with NSA can gradually add or

migrate to SA over time. At some point, of course, NSA and SA will converge as MNOs move to a full 5G architecture.

NSA

NSA refers to the joint networking of 5G and 4G LTE, which means that 5G networks are deployed based on existing 4G equipment. In the 5G NSA networking mode, operators can share 4G and 5G core networks, save network investment, but the disadvantage is that it cannot support the most core new features of 5G such as low latency. Moreover, mobile phones need to be connected to 4G and 5G networks at the same time under the NSA network (Figure 2), and the power consumption is also higher than that under the SA network. To save costs, some operators will adopt the NSA method in the early stage of 5G network construction.

In the NSA architecture, the NG-RAN and NR interface is used in combination with the existing 4G AN and its CN. The BS of the 5G NR (gNB) links to eNB through the X2 interface. The gNB is responsible for connecting the User Equipment (UE – or mobile) and the 5GC interface [3]. The S1 interface connects the eNB to the EPC made up of the Mobility Management Entity/Serving Gateway components (MME/S-GW), while the S1-U interface connects the gNB to the EPC, gNB is connected to the other gNB through the X2-U interface (Figure 3).

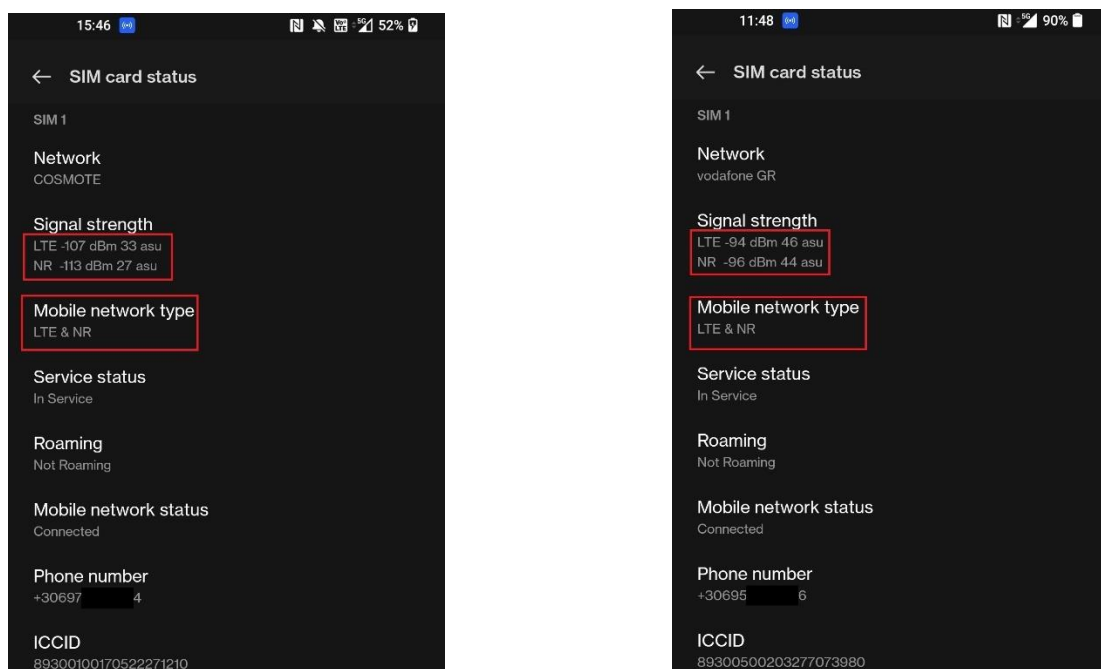


Figure 2 - Cosmote 5G NSA mode (left) & Vodafone 5G NSA mode (right)

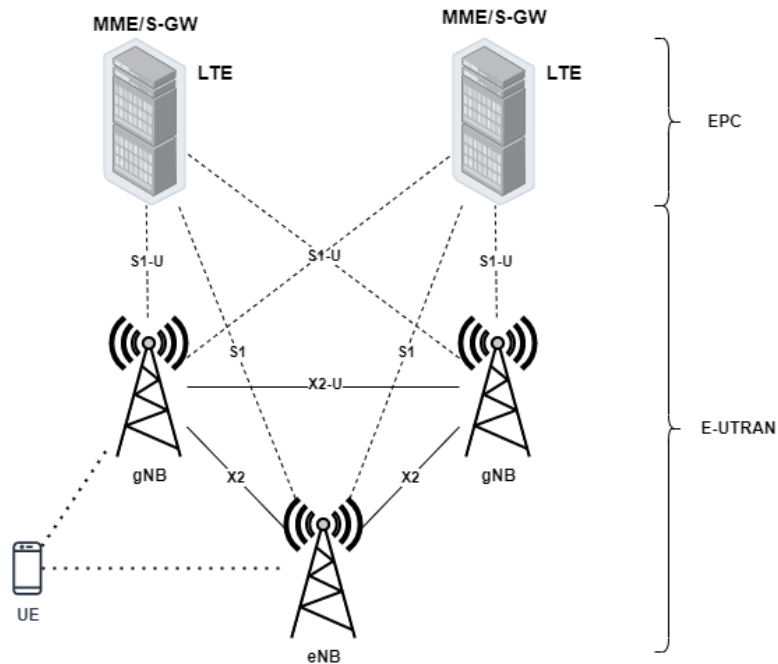


Figure 3 - 5G NSA architecture

The Evolved-UMTS Terrestrial Radio Access Network (E-UTRAN) receives information from MME and configuration data via the local E-UTRAN about the UE to determine whether to use dual connectivity or not for the UE. For the NSA, the 4G components will handle the control plane (CP) of the early 5G networks.

4G architecture

The LTE architecture consists of three main components, the UE, the EPC and the E-UTRAN (Figure 4) [4]. The **EPC** has several components, the **SGW** which routes the traffic between the base station and the Packet Data Network Gateway (PGW). The **PGW** communicates with entities outside the EPC. The Home Subscriber Server (**HSS**), which is a database server with information about subscribers, and specifically Universal Subscriber Identity Module (USIM) card records like Integrated Circuit Card Identifier (ICCID), International Mobile Subscriber Identity (IMSI). The **MME** is the main signaling node in the EPC. The Policy Control and Charging Rules Function (PCRF) is a component which is not shown in the diagram below, but it is responsible for policy control decision-making, as well as for controlling the flow-based charging

functionalities in the Policy Control Enforcement Function (PCEF), which resides in the PGW.

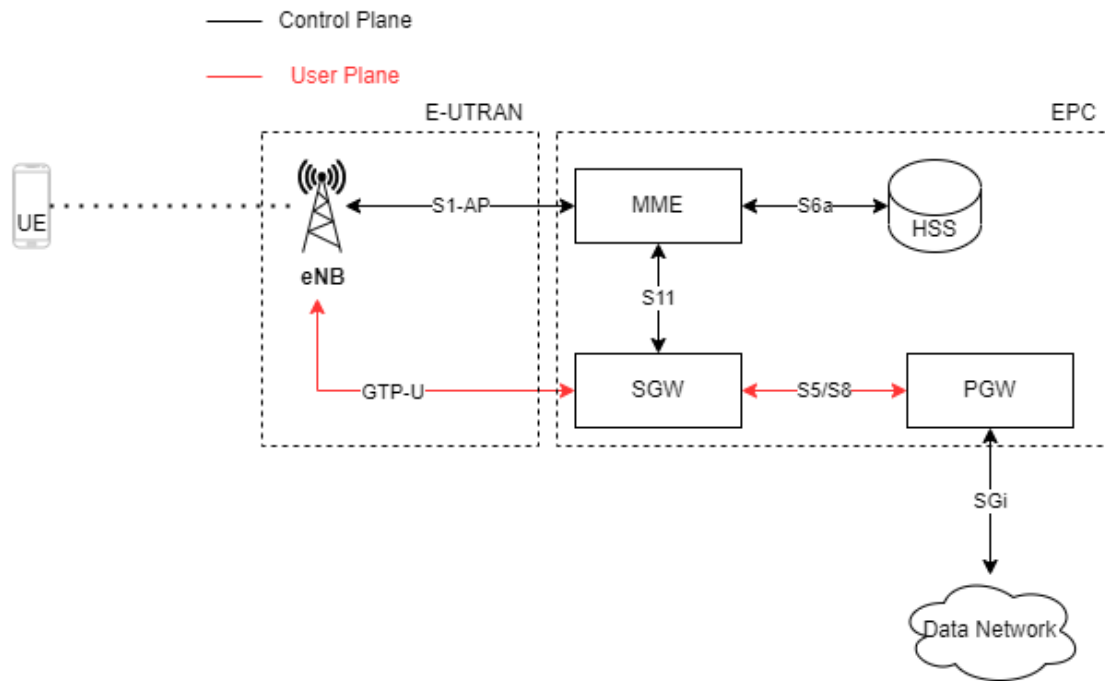


Figure 4- 4G architecture

SA

SA (Standalone) is the final form of 5G. Infrastructure such as base stations, backhaul links, and core networks are all dedicated to 5G, which can maximize the potential of 5G networks. For most operators, the difficulty of SA construction and capital investment are the biggest flaws. Therefore, many countries around the world are slowly transitioning from NSA to SA.

The 5G-SA architecture can be partitioned into the following three main components: UE, 5G-RAN and the 5G-CN [5] (Figure 5).

UE represents 5G system connectivity capability of the end user's device such as smartphone, wireless modem, smart meter, a sensor in the factory, or any other 5G-capable device equipped with a USIM. Each USIM is uniquely identified by its Subscription Permanent Identifier (SUPI), similarly to how previous generations (3G and 4G) USIMs were identified by their IMSI. A significant difference between 4G and 5G USIMs is that the new 5G USIMs can generate random nonces.

5G-RAN: In 5G, a geographical area is partitioned into hexagonal cells, where each cell is serviced by a gNB (5G base-station). 5G cells are served by low-power antennas which cover smaller geographical areas than 4G, however they provide higher data rates, and lower latency. The gNB is the link between the UE and 5G core network.

5G-CN: The 5G core network architecture standardized by 3GPP, enables support for increased throughput demand, reduced latency and increased reliability as per requirements of various applications and services that 5G must support. The new 5G core, as defined by 3GPP, utilizes cloud-aligned, service-based architecture (SBA) that spans across all 5G functions and interactions including authentication, security, session management and aggregation of traffic from end devices. The 5G core further emphasizes NFV as an integral design concept with virtualized software functions capable of being deployed in the network.

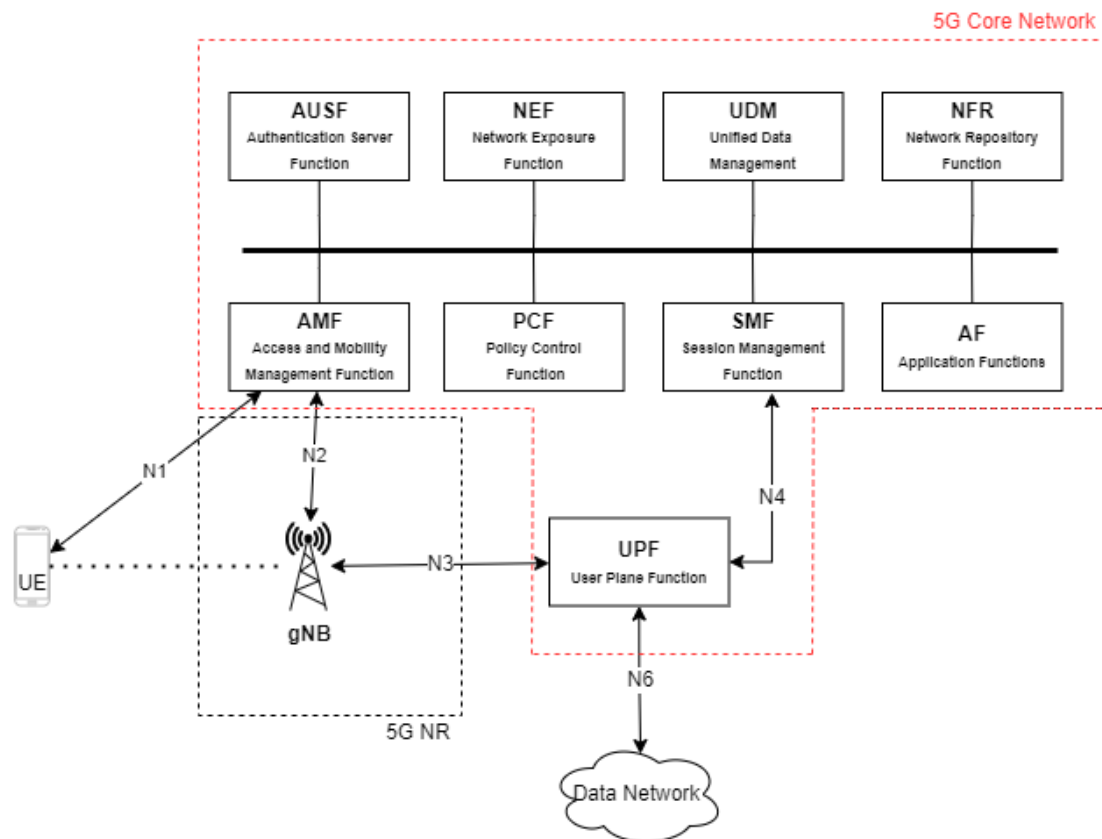


Figure 5- 5G architecture

The primary Network Functions (NFs) and their capabilities as they are defined in the standards process today are as below:

- Authentication Server Function (AUSF): Acts as an authentication server. It contains mainly the extensible authentication protocol (EAP) server

functionality and acts as storage for keys and provides keying material to the requester NF.

- Access and Mobility Management Function (AMF): It carries out termination of Non-Access Stratum (NAS) signaling, NAS ciphering and integrity protection, registration management, connection management, mobility management, access authentication and authorization, security context management. The AMF also includes the Network Slice Selection Function (NSSF) and acts as the termination point for radio access network CP interfaces (N2) (Figure 5).
- Session Management Function (SMF): It carries out session management (session establishment, modification and release), UE IP address allocation and management, Dynamic Host Configuration Protocol (DHCP) functions, termination of NAS signaling related to session management, downlink data notification and traffic steering configuration for UPF for proper traffic routing.
- User Plane Function (UPF): It carries out packet routing and forwarding, packet inspection, Quality of Service (QoS) handling, acts as external protocol data unit (PDU) session point of interconnect to Data Network (DN), and is an anchor point for Inter-Radio Access Technology mobility (RAT).
- Network Exposure Function (NEF): It supports exposure of capabilities and events, secure provision of information from external application to 3GPP network and translation of internal/external information. It acts as an API gateway that allows external users, such as enterprises or partner operators, the ability to monitor, provision and enforce application policy, for users inside the operator network. Thus, it
 - Provides security when services or Application Functions (AF) access 5G Core nodes
 - Acts as a proxy, or API aggregation point, or translator into the Core Network
- NF Repository Function (NRF): The network repository function (NRF) discovers network function instances. When it receives an NF discovery request from a NF instance, it provides the discovered NF instances. It is not present in 4G. It maintains/supports
 - a. Profiles of Network Function (NF) instances and their supported services within the network

- b. Service-Based Interfaces, Management & Maintenance
- Policy Control Function (PCF): It carries out a unified policy framework, providing policy rules to CP functions, and access subscription information for policy decisions in Unified Data Repository (UDR). This provides a policy framework incorporating network slicing, roaming and mobility management.
- Unified Data Management (UDM): It stores subscriber data and profiles and carries out generation of Authentication and Key Agreement (AKA) credentials, user identification handling, access authorization, subscription management.
- Application Functions (AF): The AF resembles an application server that can interact with the other control-plane NFs. AFs can exist for different application services and can be owned by the network operator or by trusted third parties. For instance, the AF of an over-the-top application provider can influence routing, steering its traffic towards its external edge servers. For services considered to be trusted by the operator, the AF can access Network Functions directly whereas untrusted or third-party AFs would access the Network Functions through the NEF.

5G NR Radio Protocol Stack

LTE protocol stack is being taken as the base line for the development of 5G-NR, so there are remarkable similarities between those two. [6] 5G-NR User plane contains physical access (PHY), medium access control (MAC), radio link control (RLC), and packet data convergence protocol (PDCP) (Figure 6) same as LTE and has introduced a new layer named as service data adaptation protocol (SDAP).

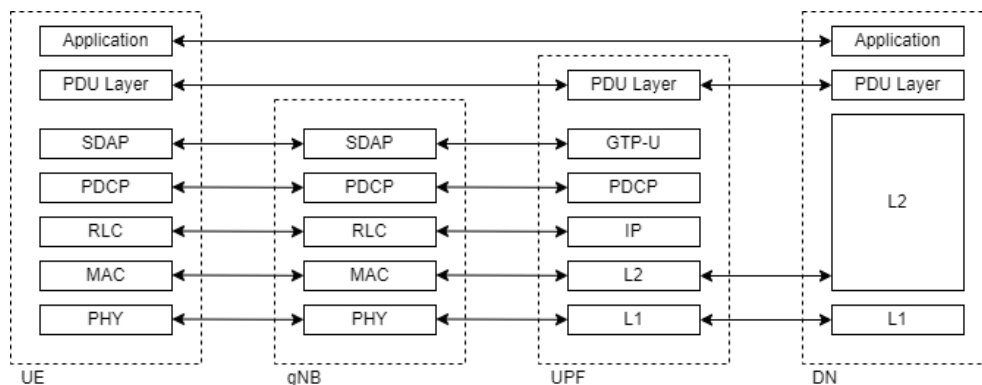


Figure 6 - NR Protocol Stack User Plane

On another side, the control plane of 5G-NR is identical to LTE, here MME equivalent node named as AMF (Figure 7).

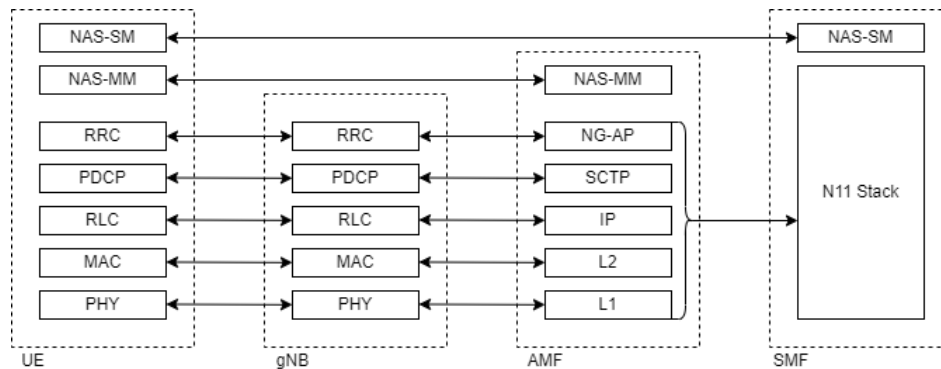


Figure 7 - NR Protocol Stack Control Plane

5G-NR Layer 2 Functions:

The layer 2 of NR is split into the following sub layers [7]:

- **Service Data Adaptation Protocol (SDAP):** The main services and functions of SDAP include Mapping between a QoS flow and a data radio bearer, marking QoS flow identification in both downlink and uplink packets. A single protocol entity of SDAP is configured for each individual PDU session, except for dual connectivity where two entities can be configured.
- **Packet Data Convergence Protocol (PDCP):** The main services and functions of the PDCP sublayer for the user plane include sequence numbering, header compression and decompression, transfer of user data, reordering and duplicate detection, PDCP PDU routing, ciphering and deciphering, PDCP service data unit (SDU) discard, PDCP re-establishment and data recovery for RLC-AM, duplication of PDCP PDUs. The main services and functions of the PDCP sublayer for the control plane include sequence numbering, ciphering, deciphering and integrity protection, transfer of control plane data, duplicate detection and duplication of PDCP PDUs.
- **Radio Link Control (RLC):** The main services and functions of the RLC sublayer depend on the transmission mode and include transfer of upper layer PDUs, sequence numbering independent of the one in PDCP, error correction through automatic repeat request (ARQ), segmentation and re-segmentation, reassembly of SDU, RLC SDU discard and RLC re-establishment.
- **and Medium Access Control (MAC):** The main services and functions of the MAC sub layer include mapping between logical channels and transport channels,

multiplexing/demultiplexing of MAC SDUs belonging to one or different logical channels into transport blocks (TB) delivered to the physical layer on transport channels, scheduling information reporting error correction through Hybrid automatic repeat request (HARQ), priority handling between UEs by means of dynamic scheduling, priority handling between logical channels of one UE by means of logical channel prioritization and padding. A single MAC entity can support one or multiple numerologies and TTI durations and mapping restrictions in logical channel prioritization controls which numerology and TTI duration a logical channel can use.

The RRC protocol is used on the Air Interface. The major functions of the RRC protocol include connection establishment and release functions, broadcast of system information, radio bearer establishment, reconfiguration and release, RRC connection mobility procedures, paging notification and release and outer loop power control. By means of the signaling functions, the RRC configures the user and control planes according to the network status and allows for Radio Resource Management strategies to be implemented.

5G-NR Layer 3 (RRC) Functions:

The main services and functions of the radio resource control (RRC) sub layer include:

- Broadcast of System Information related to Access Stratum (AS) and NAS.
- Paging initiated by 5GC or NG-RAN.
- Establishment, maintenance, and release of an RRC connection between the UE and NG-RAN including addition, modification, and release of carrier aggregation, Addition, modification, and release of Dual Connectivity in NR or between E-UTRA and NR.
- Security functions including key management.
- Establishment, configuration, maintenance, and release of Signaling Radio Bearers (SRBs) and Data Radio Bearers (DRBs).
- Mobility functions including Handover and context transfer; UE cell selection and reselection and control of cell selection and reselection; Inter-RAT mobility.
- QoS management functions.
- UE measurement reporting and control of the reporting.
- Detection of and recovery from radio link failure.
- NAS message transfer to/from NAS from/to UE.

The RRC transfers messages of the Non-Access Stratum (**NAS**), which is located above the RRC layer. NAS Mobility Management (NAS-MM) procedures are responsible to keep track of the whereabouts of the UE, UE authentication and control integrity protection and ciphering. The 5GMM procedures are also used by the network to allocate temporary identities to the UE such as 5G Global Unique Temporary Identifier (5G-GUTI) and request identity information such as subscriber concealed identifier (SUCI) and permanent equipment identifier (PEI) from the UE. In addition, the 5GMM procedures provide the UE's capability information to the network and the network may also inform the UE about information regarding specific services in the network.

5G SA Call Flow

In this section the registration call flow for over the air (OTA) messages between UE and gNB and new generation application protocol (NGAP) messages between gNB and 5GC AMF will be reviewed [5], [8], [9]. At high level, 5G SA Registration call flow includes following steps (Figure 8):

- Achieving downlink/uplink synchronization via Synchronization Signal Block (SSB) decode and RACH procedure
- SRB0 establishment with RRC connection request
- Contention resolution and SRB1 establishment with RRC setup
- Registration request
- NAS Procedures like UE Identity transfer, Authentication and Security
- AS UE Capability transfer and AS security
- SRB2 and DRB establishment
- Registration Complete and PDU session Establishment

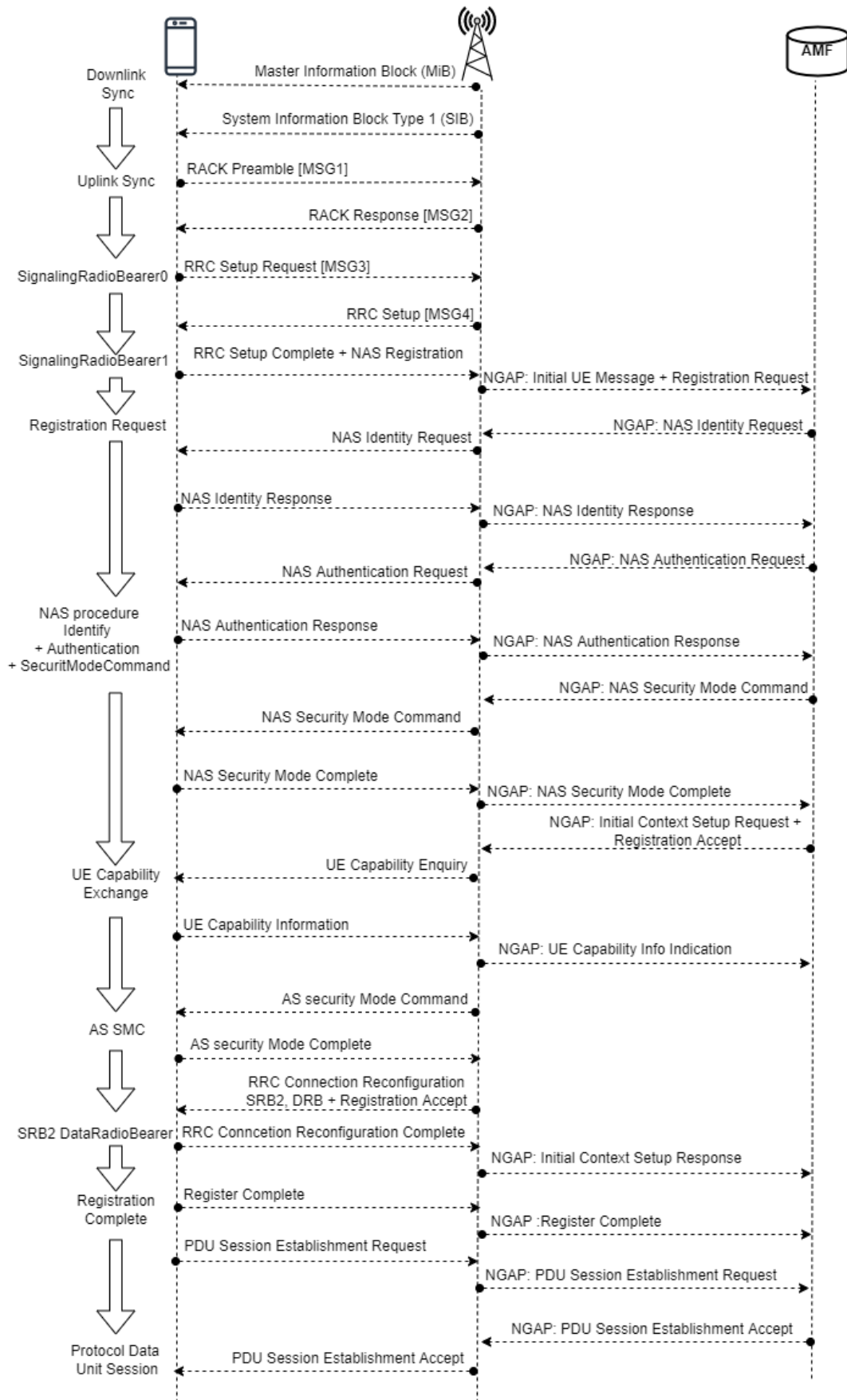


Figure 8- 5G SA register call procedure

1. **Cell Search and Downlink Synchronization:** Cell search is the procedure by which a UE acquires time and frequency synchronization with a cell, decodes the Cell ID and the Physical Broadcast Channel (PBCH) master information block (MIB) information. The process of downlink synchronization by which a UE detects the radio boundary (i.e., the exact timing when a radio frame starts) and OFDM symbol boundary (i.e., the exact timing when an OFDM symbol starts). This process is done by detecting and analyzing SSB.
2. **Uplink Synchronization** is the process by which UE figures out the exact timing when it should send uplink data (i.e., PUSCH / PUCCH). Usually, a gNB is handling multiple UEs and the network has to ensure that the uplink signal from every UE should be aligned with a common receiver timer of the network. So, this involves a much more complicated process and sometimes it must adjust UE transmission timing (uplink timing) of each UE. This is called the RACH process.
 - UE can achieve uplink synchronization by RACH procedure. UE can retrieve all RACH access related parameters from SIB. For RACH procedure, UE selects a random preamble (MSG1 - Figure 8), this preamble is referenced with the Random-Access Preamble Identifier (RAPID). UE also starts a timing T300 to await the RRC Setup message from the gNB.
 - gNB detects the RACH (MSG1) and sends a downlink control information (DCI) Format 1_0 with CRC (CRC is a hash function based on binary division) scrambled by a random-access radio network temporary identifier (RA-RNTI) corresponding to the RACH transmission. This DCI contains frequency and time resource assignment, and Modulation Coding Scheme (MCS) for (MSG2 - Figure 8) Random access response (RAR) sent on PDSCH.
 - UE tries to DCI Format 1_0 with CRC scrambled by the corresponding RA-RNTI and MAC transport block in a corresponding PDSCH having RAR (MSG2) information including timing advance, uplink grant and the Temporary cell RNTI (C-RNTI) assignment.
3. **RRC Connection Request:** RRC Connection request is considered as (MSG3 - Figure 8) and it includes UE-Identity, establishment Cause. The UE-identity can be a Random number between 0 and $2^{39}-1$ and will be used for contention resolution by UE while decoding (MSG4 Figure 8) RRC connection setup. RRC Connection Request is sent on the uplink grant provided in (MSG2 - Figure 8) from the gNB and over SRB0 on uplink common control Channel (UL_CCCH).

4. RRC setup: The RRC Setup message is sent to setup SRB1, contention resolution and the master cell configuration. The UE stops the timer T300 as it has received the RRC Setup message.
5. RRC setup Complete and Registration Request: The UE sends the RRC Setup Complete message with a “Registration Request” in the dedicated NAS message. It also carries the information about selected public land mobile network identity (PLMN-ID), registered AMF and Single Network Slice Selection Assistance Information (S-NSSAI) list. Registration request also carries UE network capability information (supporting bands-encryption algorithms). The gNB selects the AMF for this session and allocates RAN UE NGAP ID to the UE. The AMF will use this ID to address the UE context on the gNB.
6. Initial UE Message: The gNB sends the Initial UE Message to the selected AMF. The message carries the “Registration Request” message received from the UE in the RRC Setup Complete message. The “RAN UE NGAP ID” and the “RRC Establishment Cause” are also included in the message.
7. UE NAS Identity Transfer: UE Identity transfer depends on the condition if there is a change in the last AMF selected by gNB and SUCI is not provided by the UE nor retrieved from the old AMF. The Identity Request procedure is initiated by AMF sending an Identity Request message to the UE requesting the SUCI. The UE responds with Identity Response including the SUCI. The UE derives the SUCI by using the provisioned public key of the Home PLMN.
8. Authentication and NAS Security: The Core network performs Authentication procedure for the UE, that is legitimate and legally authorized to get service from the network [10]. The AMF signals the selected NAS security algorithm to the UE and requests the international mobile equipment identity software version (IMEISV) from the UE as part of NAS security mode command. UE respond with completion of the NAS security procedure and contains the IMEISV in security mode complete.
9. Initial context setup request: The AMF allocates an “AMF UE NGAP ID”. The gNB will use this ID to address the UE context on the AMF. AMF sends an INITIAL CONTEXT SETUP REQUEST message to gNB to start the initial context establishment process. The message typically contains the Registration Accept NAS message. The message carries one or more PDU Session setup requests. The

message also carries the “AMF UE NGAP ID”, “UE Aggregate Maximum Bit Rate”, UE security capabilities and security key.

10. Access Stratum UE Capability Transfer and AS Security: gNB can enquire the UE capability. After receiving the capabilities from UE, gNB updates these capabilities to AMF. The gNB sends a Security Mode Command message to the UE to notify the UE to start the integrity protection and encryption process. After that, downstream encryption is started. The UE derives the key according to the integrity protection and encryption algorithm indicated by the Security Mode Command message, and then replies with Security Mode Complete message to the gNB. After that, the upstream encryption is started.
11. SRB2 and DRB establishment: The gNB issues an RRC Reconfiguration message to the UE to establish SRB2 and DRB. After the SRB2 and DRB are successfully established, the UE replies to the gNB with an RRC Reconfiguration Complete message. The gNB signals the successful setup DRB with INITIAL CONTEXT SETUP RESPONSE message to the AMF.
12. Registration Complete and PDU session Establishment: UE sends Registration Complete and PDU session establishment request to AMF. PDU session establishment resembles PDN Connectivity Request message in LTE. AMF sends a PDU SESSION RESOURCE SETUP REQUEST message to gNB carrying the list of PDU sessions that need to be established, the list of QoS Flows for each PDU session, and the quality attributes of each QoS Flow. The gNB maps the QoS Flow to the QoS Flow according to the quality attributes of the QoS Flow. gNB sends NAS PDU session establishment accept.

Authentication Key Agreement in 4G and 5G

Authentication and key management are fundamental to the security of cellular networks because they provide mutual authentication between users and the network and derive cryptographic keys to protect both signaling and user plane data. Each generation of cellular networks always defines at least one authentication method. For example, 4G defines 4G Evolved Packet System based Authentication and Key Agreement (EPS-AKA), and 5G defines three authentication methods—5G-AKA, EAP-AKA', and EAP-TLS.

4G Authentication

Each UE has a universal integrated circuit card (UICC) hosting at least a USIM application, which stores a cryptographic key that is shared with the subscriber's home network. A serving network in 4G consists of radio access equipment such as an Evolved NodeB (eNodeB) base station and MMEs, among others. The UE communicates with a serving network through radio interfaces. A home network in 4G usually consists of authentication servers such as the HSS, which stores user credentials and authenticates users. Communication between serving networks and a home network is based on IP; the core entities that are connected over an IP network are collectively referred to as the Evolved Packet System (EPS).

4G EPS-AKA

The EPS-AKA is triggered after the UE completes the RRC procedure with eNodeB and sends an Attach Request message to the MME (Figure 9). The MME sends an Authentication Request, including UE identity (IMSI) and the serving network identifier, to the HSS located in the home network. The HSS performs cryptographic operations based on the shared secret key, K_i (shared with the UE), to derive one or more authentication vectors (AVs), which are sent back to the MME in an Authentication Response message. An AV consists of an authentication (AUTH) token and an expected authentication response (XAUTH) token, among other data.

After receiving an Authentication Response message from the HSS, the MME sends an Authentication Request to the UE, including the AUTH token. The UE validates the AUTH token by comparing it to a generated token based on K_i . If the validation succeeds, the UE considers the network to be legitimate and sends an Authentication Response message back to the MME, including a response (RES) token, which is also generated based on K_i .

The MME compares the RES token with an expected response (XRES) token. If they are equal, the MME performs key derivation and sends a Security Mode Command message to the UE, which then derives the corresponding keys for protecting subsequent NAS signaling messages. The MME will also send the eNodeB a key from which the keys for protecting the RRC channel are derived. After the UE also derives

the corresponding keys, subsequent communication between the UE and the eNodeB is then protected.

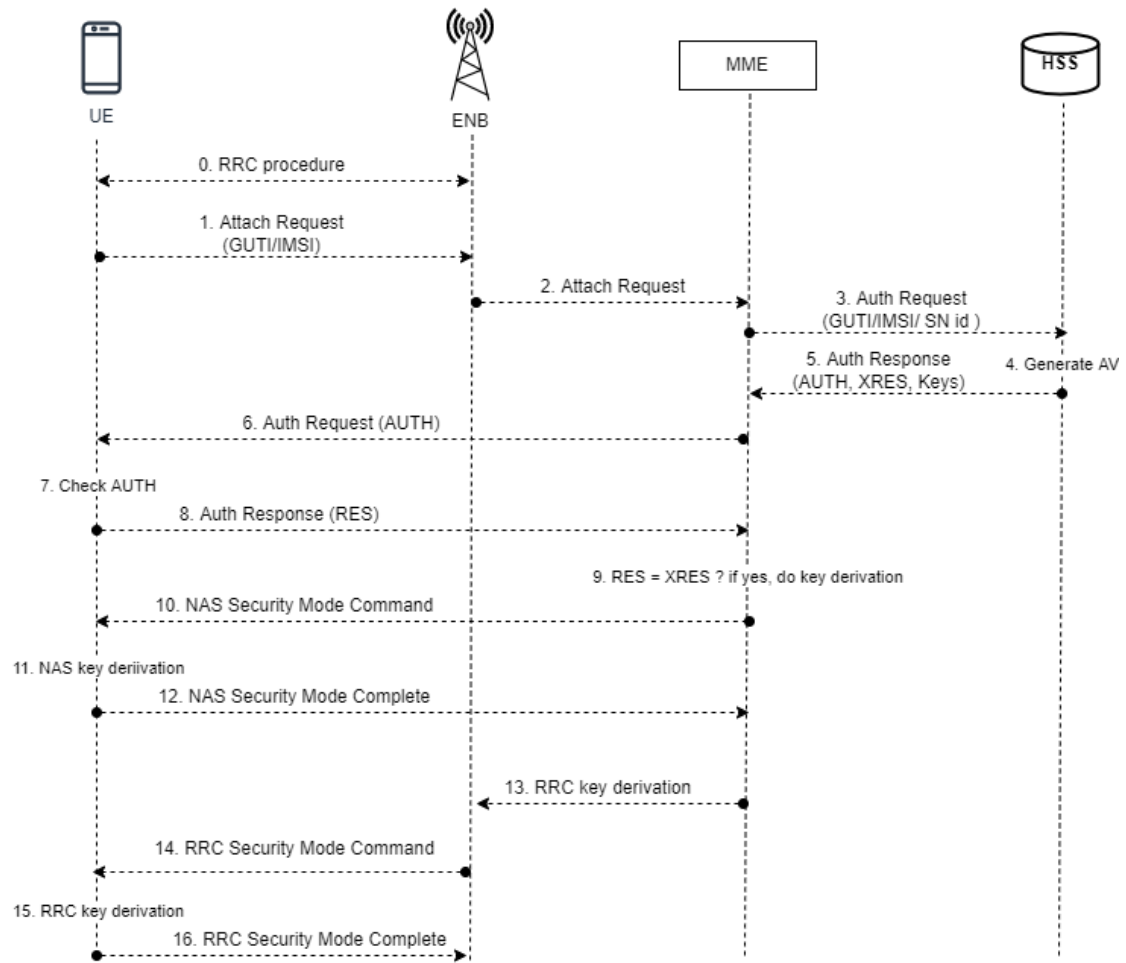


Figure 9- 4G Authentication Procedure

5G Authentication

SBA has been proposed for the 5G core network. Accordingly, new entities and new service requests have also been defined in 5G. Some of the new entities relevant to 5G authentication are listed below.

- The Security Anchor Function (SEAF) is in a serving network and is a “middleman” during the authentication process between a UE and its home network. It can reject an authentication from the UE, but it relies on the UE’s home network to accept the authentication.
- The AUSF is in a home network and performs authentication with a UE. It makes the decision on UE authentication, but it relies on backend service for

computing the authentication data and keying materials when 5G-AKA or EAP-AKA' is used.

- Unified data management (UDM) is an entity that hosts functions related to data management, such as the Authentication Credential Repository and Processing Function (ARPF), which selects an authentication method based on subscriber identity and configured policy and computes the authentication data and keying materials for the AUSF if needed.
- The Subscription Identifier De-concealing Function (SIDF) decrypts a Subscription Concealed Identifier (SUCI) to obtain its long-term identity, namely SUPI [11]. In 5G, a subscriber's long-term identity is always transmitted over the radio interfaces in an encrypted form. More specifically, a public key-based encryption is used to protect the SUPI. Therefore, only the SIDF has access to the private key associated with a public key distributed to UEs for encrypting their SUPIs.

5G Authentication Framework

A unified authentication framework has been defined to make 5G authentication both open (e.g., with the support of EAP) and access-network agnostic (e.g., supporting both 3GPP access networks and non-3GPP access networks such as Wi-Fi and cable networks) (Figure 10). When EAP is used (e.g., EAP-AKA' or EAP-TLS), EAP authentication is between the UE and the AUSF through the SEAF (functioning as an EAP pass-through authenticator). When authentication is over untrusted, non-3GPP access networks, a new entity, namely the Non-3GPP Interworking Function (N3IWF), is required to function as a VPN server to allow the UE to access the 5G core over untrusted, non-3GPP networks through IPsec tunnels. Several security contexts can be established with one authentication execution, allowing the UE to move from a 3GPP access network to a non-3GPP network without having to be re-authenticated.

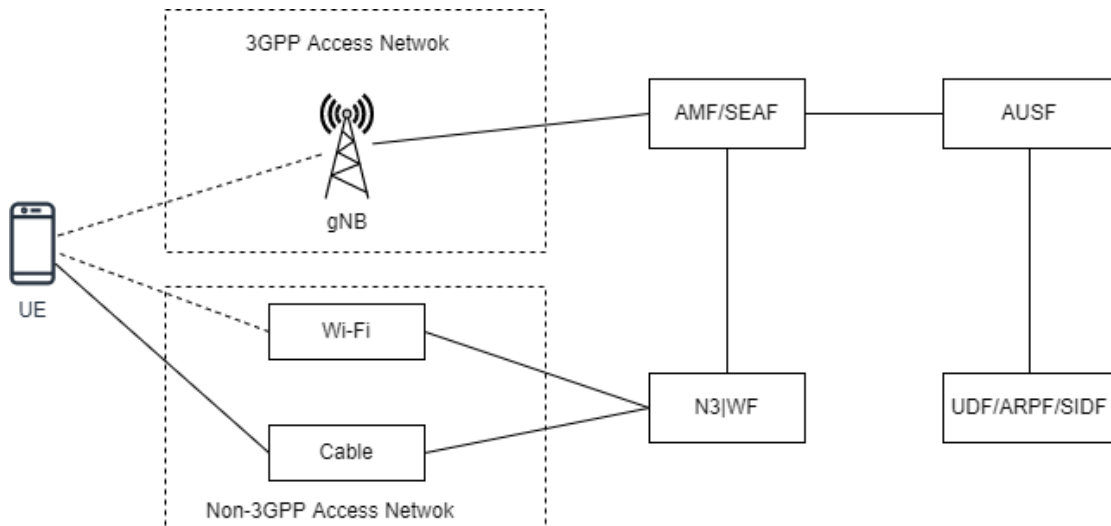


Figure 10- 5G Authentication Framework

5G-AKA

5G defines new authentication-related services. For simplicity, generic messages such as Authentication Request and Authentication Response are used (Figure 11) without referring to the actual authentication service names. In 5G-AKA, the SEAF starts the authentication procedure after receiving any signaling message from the UE. A UE should send the SEAF a temporary identifier (a 5G-GUTI) or a SUCI if a 5G-GUTI has not been allocated by the serving network for the UE. The SEAF starts the authentication by sending a request to the AUSF, which first verifies that the serving network requesting the authentication service is authorized. Upon success, the AUSF sends an authentication request to UDM/ARPF. If a SUCI is provided by the AUSF, then the SIDF will be invoked to decrypt the SUCI to obtain the SUPI, which is further used to select the authentication method configured for the subscriber. In this case, it is 5G-AKA, which is selected and to be executed. UDM/ARPF starts 5G-AKA by sending the authentication response to the AUSF with an authentication vector consisting of an AUTH token, an XRES token, the key K_{AUSF} , and the SUPI if applicable (e.g., when a SUCI is included in the corresponding authentication request), among other data.

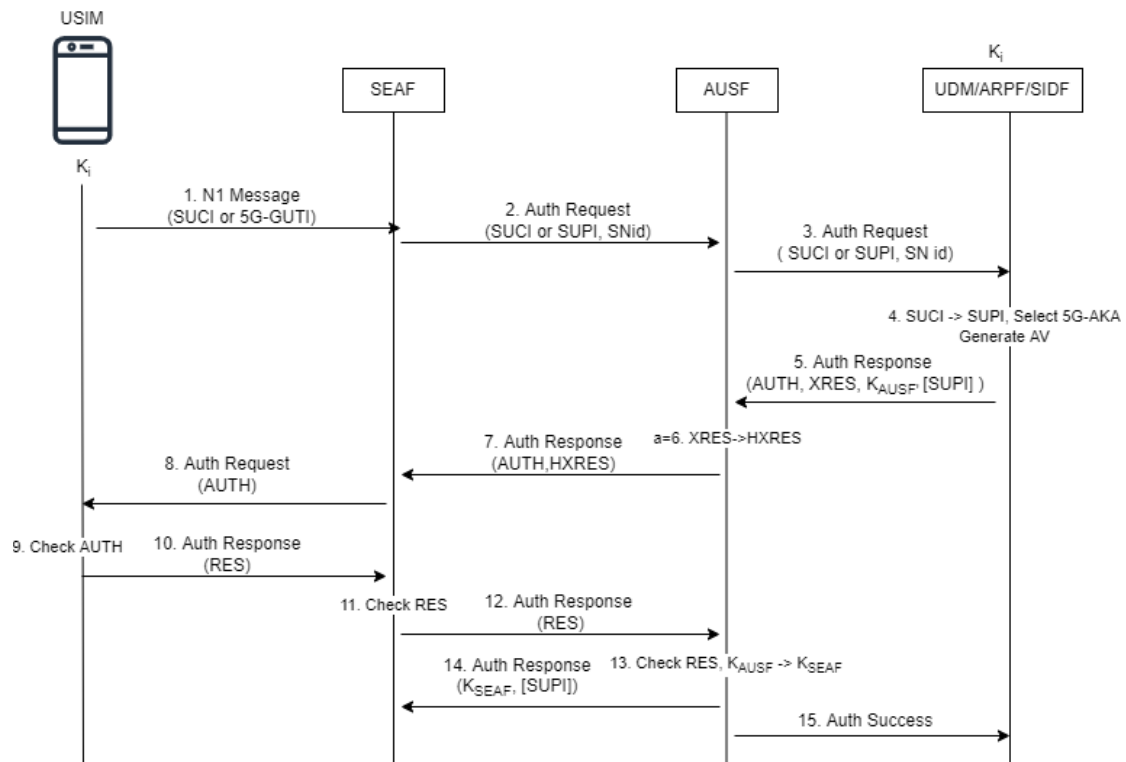


Figure 11- 5G Authentication Procedure

The AUSF computes a hash of the expected response token (HXRES), stores the K_{AUSF} , and sends the authentication response to the SEAF, along with the AUTH token and the HXRES. Note that the SUPI is not sent to the SEAF in this authentication response. It is only sent to the SEAF after UE authentication succeeds. The SEAF stores the HXRES and sends the AUTH token in an authentication request to the UE. The UE validates the AUTH token by using the secret key it shares with the home network. If validation succeeds, the UE considers the network to be authenticated. The UE continues the authentication by computing and sending the SEAF a RES token, which is validated by the SEAF. Upon success, the RES token is further sent by the SEAF to the AUSF for validation. Note that the AUSF, which is in a home network, makes the final decision on authentication. If the RES token from the UE is valid, the AUSF computes an anchor key (K_{SEAF}) and sends it to the SEAF, along with the SUPI if applicable. The AUSF also informs UDM/ARPF of the authentication results so they can log the events, e.g., for the purpose of auditing. Upon receiving the K_{SEAF} , the SEAF derives the AMF key (K_{AMF}) (and then deletes the K_{SEAF} immediately) and sends the K_{AMF} to the co-located Access and Mobility Management Function (AMF). The AMF will then derive from the K_{AMF} (a) the confidentiality and integrity keys needed to

protect signaling messages between the UE and the AMF and (b) another key, K_{gNB} , which is sent to the gNB base station for deriving the keys used to protect subsequent communication between the UE and the gNB. Note that the UE has the long-term key, which is the root of the key derivation hierarchy. Thus, the UE can derive all above keys, resulting in a shared set of keys between the UE and the network.

EAP-AKA'

EAP-AKA' [11] is another authentication method supported in 5G. It is also a challenge-and-response protocol based on a cryptographic key shared between a UE and its home network. It accomplishes the same level of security properties as 5G-AKA, e.g., mutual authentication between the UE and the network. Because it is based on EAP, its message flows differ from those of 5G-AKA. Note that EAP messages are encapsulated in NAS messages between the UE and the SEAF and in 5G service messages between the SEAF and the AUSF. Other differences between 5G-AKA and EAP-AKA' are:

- In EAP-AKA', EAP message exchanges are between the UE and the AUSF through the SEAF, which transparently forwards the EAP messages without being involved in any authentication decision. In 5G-AKA, the SEAF also verifies the authentication response from the UE and may act if the verification fails, albeit such action has not yet been defined in 3GPP [12].
- Key derivation differs slightly. In 5G-AKA, the K_{AUSF} is computed by UDM/ARPF and sent to the AUSF. In EAP-AKA', the AUSF derives the K_{AUSF} itself in part based on the keying materials received from UDM/ARPF. More specifically, the AUSF derives an Extended Master Session Key (EMSK) based on the keying materials received from UDM according to EAP and then uses the first 256 bits of the EMSK as the K_{AUSF} .

EAP-TLS

EAP-TLS [13] is defined in 5G for subscriber authentication in limited use cases such as private networks and IoT environments. When selected as the authentication method

by UDM/ARPF, EAP-TLS is performed between the UE and the AUSF through the SEAF, which functions as a transparent EAP authenticator by forwarding EAP-TLS messages between the UE and the AUSF. To accomplish mutual authentication, both the UE and the AUSF can verify each other's certificate or a pre-shared key (PSK) if it has been established in a prior Transport Layer Security (TLS) handshake or out of band. At the end of EAP-TLS, an EMSK is derived, and the first 256 bits of the EMSK is used as the K_{AUSF} . As in 5G-AKA and EAP-AKA', the K_{AUSF} is used to derive the K_{SEAF} , which is further used to derive other keying materials (Figure 12) needed to protect communication between the UE and the network. EAP-TLS fundamentally differs from 5G-AKA and EAP-AKA' in its trust establishment between a UE and the network, i.e., it uses a different trust model. In EAP-TLS, mutual authentication between a UE and a 5G network is obtained primarily based on the mutual trust of their public key certificates, acknowledging that TLS with a PSK is possible but is rarely used except for session resumption. In AKA-based methods, such trust is based solely on a symmetric key shared between a UE and the network. Such a fundamental difference is significant in that EAP-TLS removes the need to store a large number of long-term keys in the home network, thus reducing operational risks in the life cycle of symmetric key management. On the other hand, EAP-TLS introduces new overhead in certificate management, such as certificate issuance and revocation.

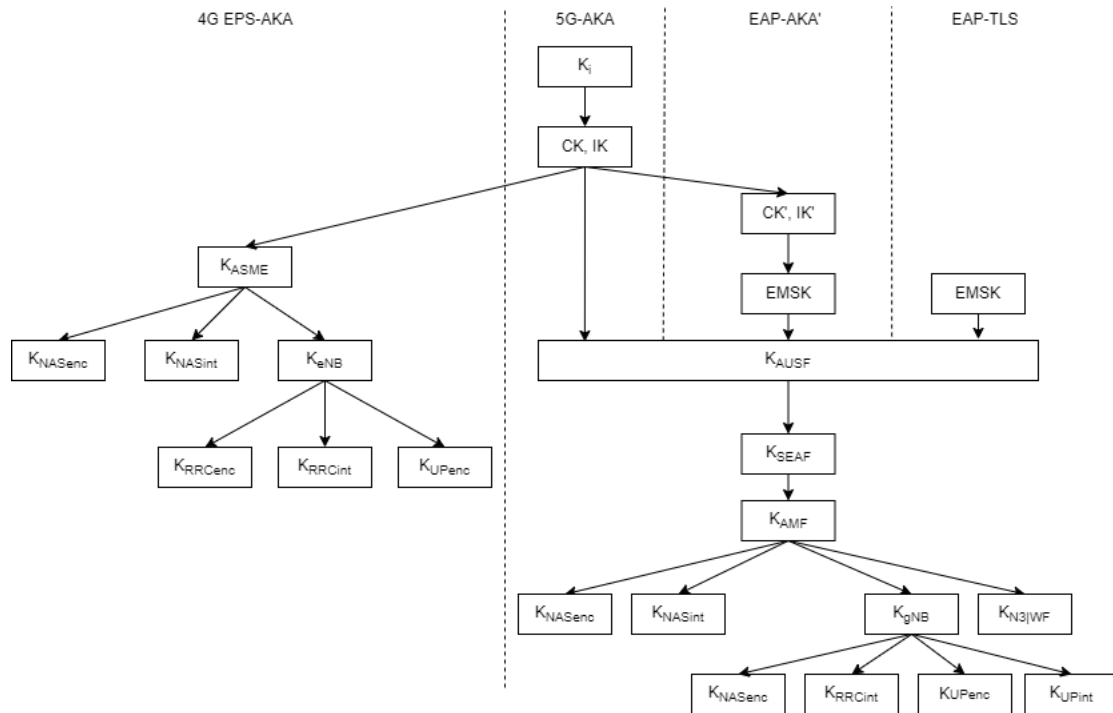


Figure 12- Key Hierarchy in 4G and 5G

Subscriber Privacy

A major evolution in the 5G system is the protection of the subscriber privacy. With subscriber privacy mechanism, the SUPI can be sent on the radio interface encrypted. The SUPI in 5G is the equivalent of the IMSI in 4G. To counteract privacy-related attacks such as false base station attack, the user equipment computes and sends the SUCI: a privacy preserving identifier containing the concealed SUPI. However, sending the SUCI instead of the SUPI is not mandatory, the use of the subscriber privacy mechanism depends on the home network operator decision taking into national regulations. Thanks to the USIM configuration chosen by the home network operator, the UE knows whether a SUCI must be sent rather than the SUPI and whether the computation of the SUCI takes place in the USIM or in the UE. The Home Network Public Key used to conceal the SUPI is always provisioned and stored only in the USIM.

- If the operator's decision is that UE must calculate the SUCI, then the home network operator provisions in the USIM an ordered priority list of the protection scheme identifiers that the operator allows. The priority list of protection scheme identifiers in the USIM only contains protection scheme identifiers as specified in Annex C [12]. This list may contain one or more protection schemes identifiers. The UE reads the SUCI calculation information from the USIM, including the SUPI, the SUPI Type, the Routing Indicator, the Home Network Public Key Identifier, the Home Network Public Key, and the list of protection scheme identifiers. The UE selects the protection scheme from its supported schemes that has the highest priority in the list obtained from the USIM.
- If the operator's decision, indicated by the USIM, is that the USIM has to calculate the SUCI, then the USIM does not provide any parameter to the UE related to the calculation of the SUCI. The operator must choose one protection scheme to conceal the SUPI between those specified in Annex C [12]; it could also be a customized protection scheme chosen by the home network operator. Consequently, the computation of the SUCI in the USIM offers more flexibility and control for the mobile network operator to protect the identities of his subscribers.

- In case that the Home Network Public Key or the priority list are not provisioned, the UE calculates the SUCI using the null-scheme, equivalent to no subscriber privacy mechanism.

Identifiers in 5G

5G has introduced relevant improvements in terms of protection mechanisms against the catching of identifiers, i.e., the capability of intercepting identifiers through eavesdropping. In 5G networks three identifiers are important: the permanent identifier SUPI, the concealed identifier SUCI, and the temporary identifier 5G-GUTI.

A Valid **SUPI** can be either of the following: an IMSI, or a Network Access Identifier (NAI). SUPI is usually a string of 15 decimal digits and acts as the long-term identifier of an individual subscriber [14], [15]. The first three digits represent the Mobile Country Code (MCC) while the next two or three form the Mobile Network Code (MNC), which identifies the network operator (Figure 13). The length of the MNC field is a national affair. The remaining (nine or ten) digits are known as Mobile Subscriber Identification Number (MSIN) and represent the individual user of that operator. The SUPI value is provisioned in USIM and UDM/UDR function in 5G Core.

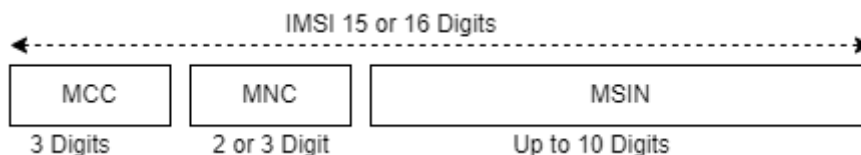


Figure 13 - SUPI structure

SUCI is a privacy preserving identifier containing the concealed SUPI. The UE generates a SUCI using an ECIES-based protection scheme with the public key of the Home Network that was securely provisioned to the USIM during the USIM registration. Only the MSIN part of the SUPI gets concealed by the protection scheme while the home network identifier i.e., MCC/MNC gets transmitted in plain text.

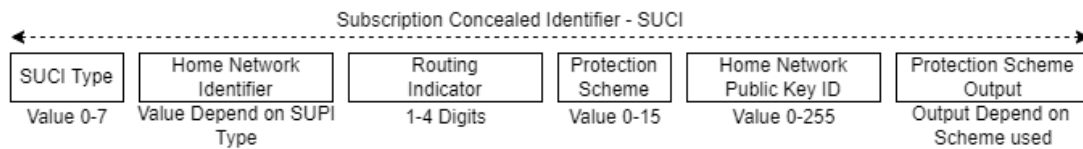


Figure 14- SUCI structure

The data fields constituting the SUCI (Figure 14) are following:

- **SUPI Type**: consisting in a value in the range 0 to 7. It identifies the type of the SUPI concealed in the SUCI. The following values are defined
 - 0: IMSI
 - 1: Network Access Identifier (NAI)
 - 2 to 7: spare values for future use.
- **Home Network Identifier**: Identifying the home network of the subscriber. When the SUPI Type is an IMSI, the Home Network Identifier is composed of MCC and MNC. When the SUPI type is a Network Access Identifier, the Home Network Identifier consists of a string of characters with a variable length representing a domain name (i.e., user@domain.com).
- **Routing Indicator**: It consists of 1 to 4 decimal digits assigned by the home network operator and provisioned within the USIM.
- **Protection Scheme Identifier**: It is consisting of a value in the range of 0 to 15 and represented with 4 bits
 - null scheme 0x0
 - Profile <A> 0x1
 - Profile 0x2
- **Home Network Public Key Identifier**: It consists of a value in the range 0 to 255. It represents a public key provisioned by the HPLMN and it is used to identify the key used for SUPI protection. In case of null scheme being used, this data field shall be set to the value as 0
- **Protection Scheme Output**: It consists of a string of characters with a variable length or hexadecimal digits, dependent on the used protection scheme

5G-GUTI is a core network temporary identifier allocated by AMF to the UE. It consists of three network identities PLMN, AMF ID, and temporary mobile subscriber identity (TMSI) (Figure 15). Its application preserves the privacy of subscribers, by not

allowing one to be associated with a permanent identifier. Moreover 5G-GUTI can be used for accessing 3GPP and non-3GPP networks.

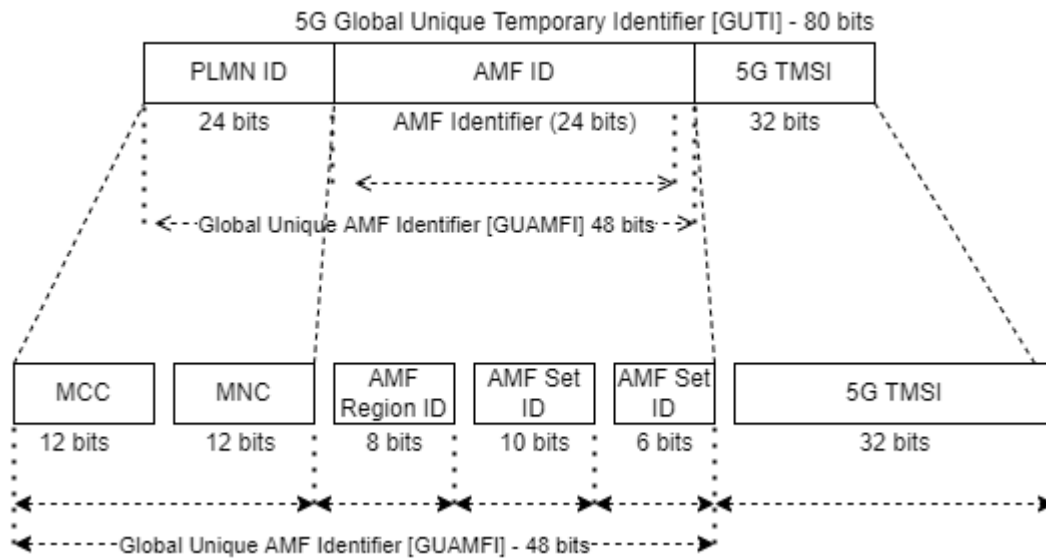


Figure 15-GUTI structure

This temporary identifier needs to change on several occasions:

- on initial registration request messages
- on periodic registration request messages
- on service request messages (i.e., after a paging request).

Evaluation of MNOs 5G deployment

As already mentioned, MNOs are faced with critical dilemmas, whether they should invest on 5G SA or 5G NSA for their consumers. While NSA deployment allows them to leverage their existing network investments in transport and mobile core, SA deployment enables them to cover market gaps, by reaching new enterprises and offering them network slicing, edge computing, and more 5G SA capabilities [16]. So, most MNOs will gradually migrate to SA over time. However, this is not the only quandary network operators are facing, as the broader ecosystem is not yet mature. Issues regarding battery optimization on mobile devices connected to 5G SA have emerged, as for silicon, LTE is more efficient from a power-draw perspective. From the end user perspective, 5G mostly offers users "more of the same" – especially with the current deployments and their re-use of existing 4G core components.

Currently, early 5G networks are NSA, some 5G networks also re-use the 4G radio network, with techniques like dual connectivity or dynamic spectrum-sharing. 5G, is still under development, whether that is using NSA or SA cores. Even where a 5G logo is displayed on a smartphone screen, the performance varies hugely depending on the spectrum bands and specific configurations involved.

Mobile Network Inspection Tools

Mobile networks are a complex system that the final user is unaware of. As mentioned before on the 5G architecture, it incorporates multiple wireless communication, mobility management, data transfer and security functions on the control plane and user plane [17]. Open-source tools have been developed to fulfill this gap between the mobile networks and the final user. QCSuper [18] is a tool, leveraging the chipset of Qualcomm-based phones and modems, allowing it to capture raw radio frames, among other things. It uses the Qualcomm Diag protocol, also called QCDDM or Diagnostic Monitor in order to communicate with mobile's baseband. However, it currently does not encapsulate 5G protocols, and the creators propel the usage of MobileInsight [19].

MobileInsight

MobileInsight is a software tool implemented for collection and analysis of runtime network information from cellular networks [20], developed by researchers at UCLA Wireless Networking Group and Purdue Peng Group, licensed under Apache License 2.0. It runs on commercial phones without additional equipment or support from operators. It exposes protocol messages on the control plane and data plane from the 4G/5G chipset. By offering a simple Application Programming Interface (API), developers can obtain access to low-level network information. It runs on user-space and supports per-message information retrieval and analysis from a set of cellular-specific protocols. It first exposes raw cellular logs from the cellular interface to the device user-space at runtime, and then parses them into protocol messages and extracts their carried information elements. It builds an extensible modular framework, where each parser works on a per-protocol basis. The parsed messages are then passed to the analyzer. The analyzer given the extracted messages, aims to unveil protocol dynamics and operation logics. Based on the observed messages and the anticipated behavior model, the analyzer infers protocol states, triggering conditions for state transitions, and protocol's actions. It offers embedded abstraction per protocol and enables developers to customize their analyzers.

Solution Approach

The team that developed MobileInsight, overcame the problem using ordinary devices that did not expose message-level cellular information to the user space, by leveraging a side channel between the chipset and the software. They used the chipset's support of an external diagnostic mode, which exposed the cellular interface to the USB port. This diagnostic mode in fact exists for major cellular chipsets (including Qualcomm, MediaTek, and Intel series) and mobile operating systems (OS). However, there is no literature freely available for this mode. The team got information about this mode from open-source code of diagnostic drivers. The cellular interface maps to a virtual device (`/dev/diag`) in the OS. Unlike the radio interface layer, this virtual device exposes all raw cellular messages as binary streams. When the USB is connected to the external collector (a PC), the OS uses USB tethering to bind the virtual device with a USB port

(/dev/ttyUSB*). The external collector fetches the cellular messages from the hardware interface.

Application Architecture

The MobileInsight core has two types of modules: monitors and analyzers. The monitor extracts low-level logs/messages from the device's network protocol stack at real-time and can be used to drive the mobile network analysis. The analyzers are event-driven and can perform online/offline mobile network analysis. MobileInsight supports analysis of major cellular protocols, including RRC, NAS, AS. The monitors can run separately without analyzers, but the analyzers should be driven by at least one monitor. MobileInsight allows the user to develop its own analyzers for customized usage. The users can build their own analyzes on top of existing ones (Figure 16). Analyzers can run concurrently for multiple analysis

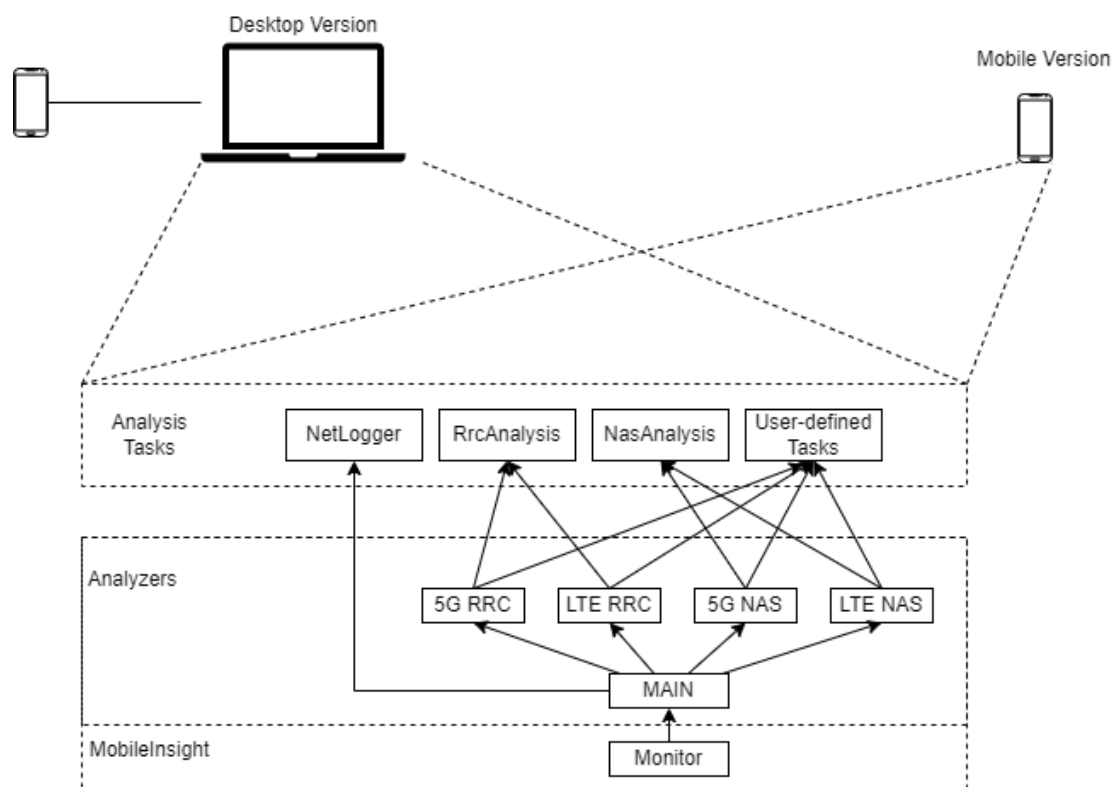


Figure 16- MobileInsight high level architecture

MobileInsight has both mobile and desktop versions. The mobile version as the name suggests is executed in the phone. It can be used to collect background logs, and benefit mobile phones in a variety of scenarios, such as failure diagnosis, performance

improvement, and security loophole detection. The desktop version supports Linux/OSX and is suitable for more complex analysis. It requires connecting the phone to a desktop device. Both the mobile and desktop version share the same APIs. The same monitoring/analysis code can run on both versions without modifications. MobileInsight supports Android phones with a Qualcomm chipset only.

Mobile Interface

The device used for testing the mobile application of MobileInsight, is a OnePlus 8 pro, with Qualcomm snapdragon 865 processor, running android 11, and later updated and tested on the same mobile on android 12. Device was granted root privileges, as the application requires. Moreover, diagnosis mode was enabled in the mobile as this is needed for the application to cooperate with Qualcomm processors. Unfortunately, this device is not on the list of the supported ones, and it malfunctioned in specific cases, such as storing the captured RRC or NAS messages, or on providing online analysis of these messages. The interface is quite well designed and easy to use. There is a central menu where users can select either a plugin, browse on captured log files or change settings (Figure 17).

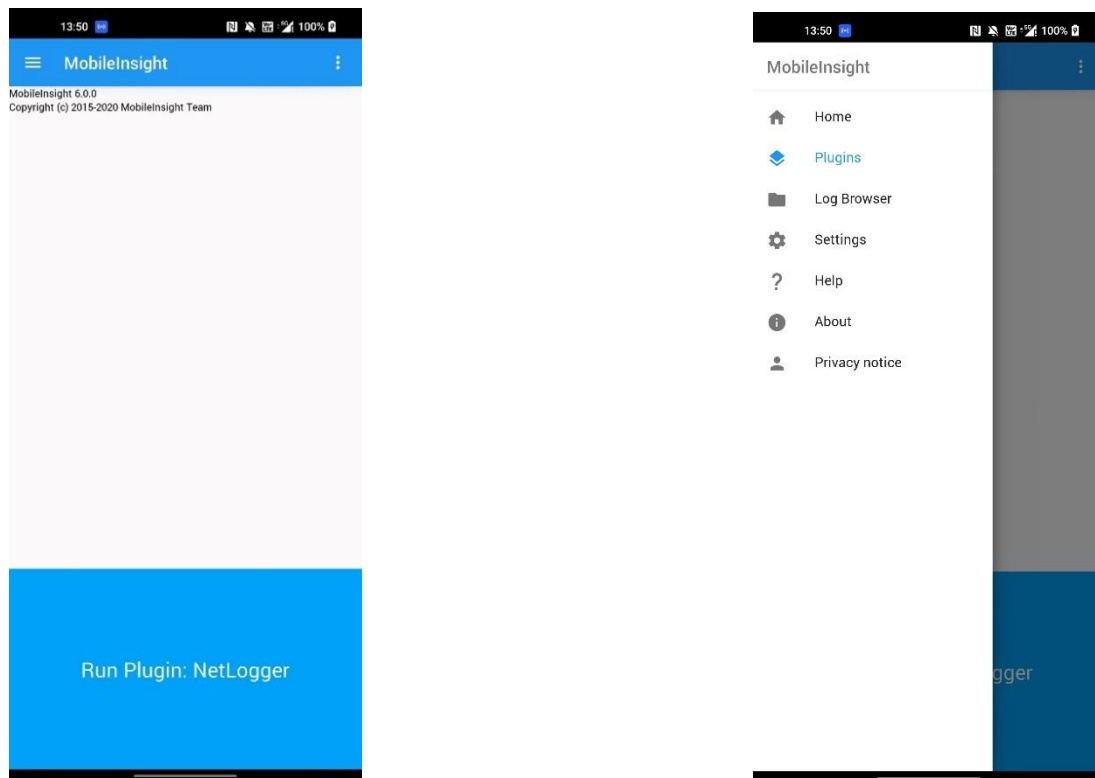


Figure 17- MobileInsight App Home (left) & Menu (right)

Having select the corresponding item from the menu, user can navigate through plugins:

- **NetLogger:** Logs cellular messages and saves them into a .mi2log file. Specific messages can be captured by customizing the configuration.
- **RrcAnalysis:** Provides an analyzer for 3G/4G/5G RRC protocols.
- **NasAnalysis:** Provides an analyzer for 3G/4G/5G session and mobility management protocols
- **WifiMonitor:** Provides analytics for Wifi information
- **KPIAnalyzer:** Provides a runtime key performance indicator for mobile networks.

By selecting settings, and then the plugin, users can change various configurations from Log Type (Figure 18) to setting size of the logs or enabling decoding of logs.

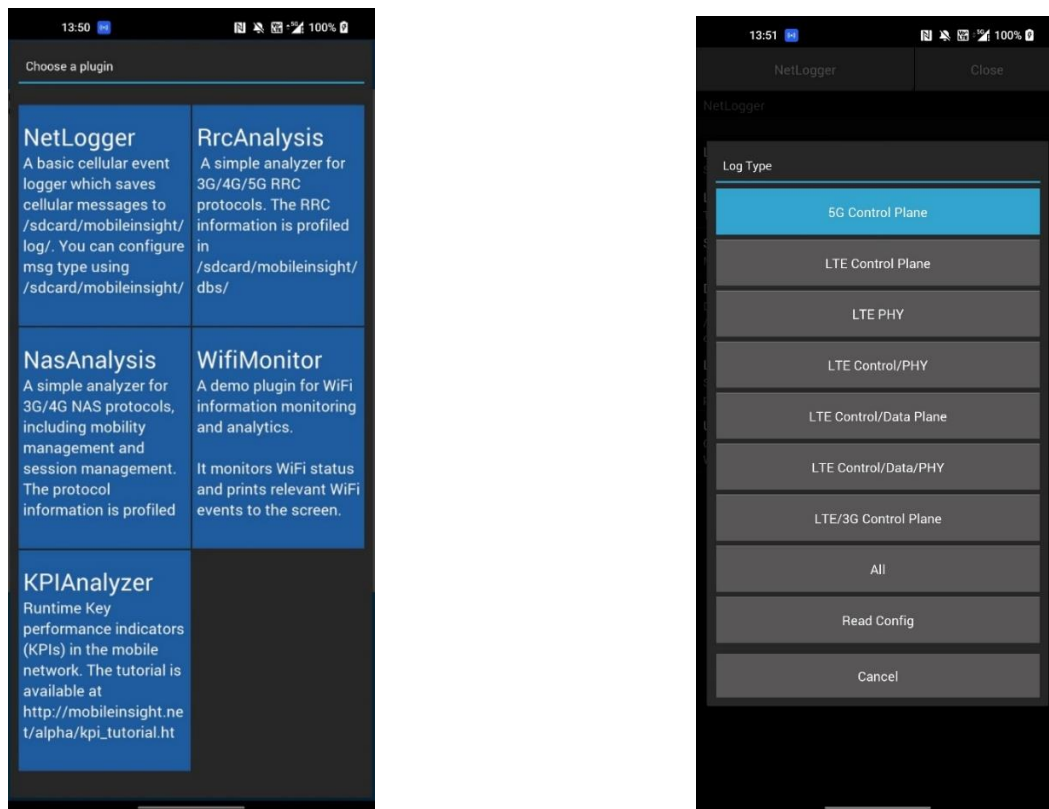


Figure 18- MobileInsight plugins (left) & LogType (right)

However, users can run their customized plugins on the phone as MobileInsight is designed to provide a platform independent environment. If an analysis script is developed with built-in monitors and analyzers only, it should run on both desktop

servers and phones without modifications. The only extra step is to wrap the code as a plugin. In the MobileInsight mobile version, all the user-defined plugins should be placed in /sdcard/mobileinsight/plugins/ path. Each plugin is placed in a separate folder, and includes at least a plugin entrance (main.mi2app), a description file (readme.txt), and an optional setting panel configuration (settings.json). Any number of files can be included, such as customized analyzers, metadata, resource files, etc. More specifically, the directory structure of plugins looks like this:

```

/sdcard/mobileinsight/plugins/
  Plugin1/
    main.mi2app
    readme.txt
    settings.json
  Plugin2/
    main.mi2app
    readme.txt
    settings.json
    __init__.py
    files/
    ...
  ...
  
```

At runtime, MobileInsight will automatically recognize the new plugin and display it, so that the user can select it and run the code.

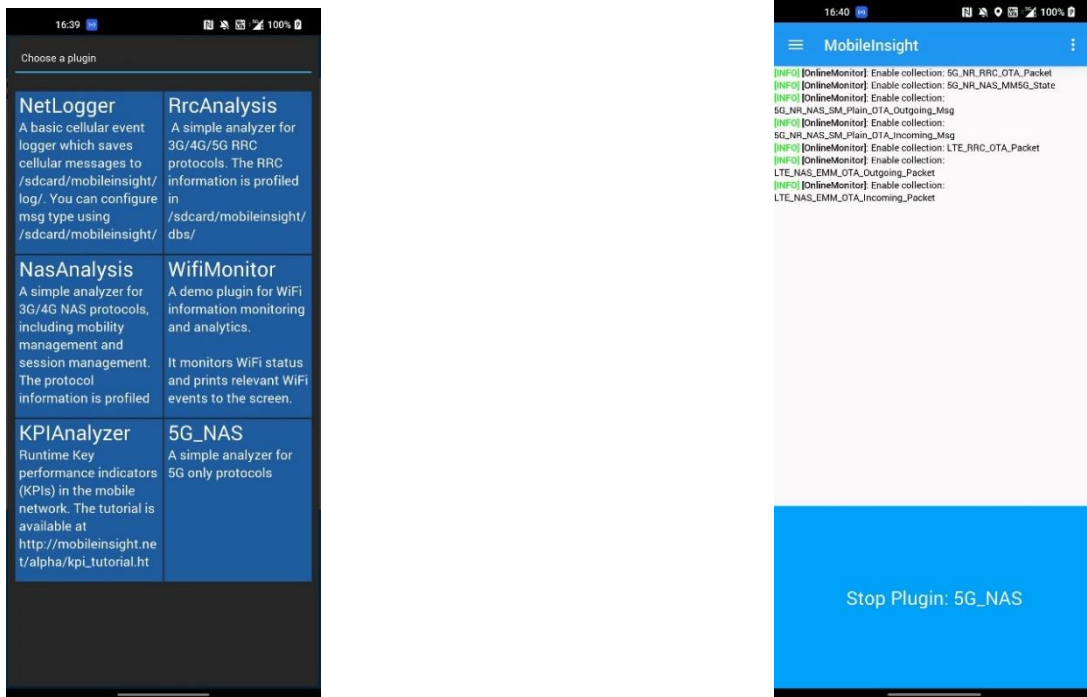


Figure 19- MobileInsight custom plugin

By creating these files mentioned above, placing them all under a directory with an arbitrary name (i.e., 5G_NAS), and by copying them in the path `/sdcard/mobileinsight/plugins/5G_NAS` a custom plugin can be created. This could help users focus on analyzing specific messages. The files used for the Plugin creation can be found in the section Custom Plugin for MobileInsight Mobile Interface.

Desktop Interface

The desktop edition of MobileInsight, is a set of python scripts, mostly used from terminal. There is a graphical user interface supporting the ease of use, but it only concerns the analysis of the captured message's part. The host PC needs to have downloaded and installed the MobileInsight. Detailed instructions are available on GitHub repo [19], and on their website [21], moreover there is a solution of a portable development environment with all tools pre-installed by using a tool named Vagrant. For this demonstration, Ubuntu 20.04 was used, with a manual installation of MobileInsight. After the installation, the mobile should be connected to the PC. The procedure for configuring the phone connection to the Desktop is complicated and some background knowledge on android debug bridge (adb) is needed. There are several steps needed to complete the connection of the mobile to the PC. First, diagnostic mode on the phone must be enabled. This procedure is phone-specific, however there is a universal way (note mobile must be rooted and chipset manufacturer Qualcomm) through adb commands (Developer options must be enabled, and USB debugging selected). After connecting mobile via USB to PC (Transfer photos or transfer files connection mode) and using the commands `adb devices` which lists available devices in debug mode, a device must appear. Then by using command `adb shell`, a shell should open which is a terminal from the mobile. By providing the command `setprop sys.usb.config diag,adb` to the mobile (Figure 20), diagnostic mode is enabled (note that after pressing enter on this command, the mobile shell will be terminated, as this command forces usb connection in another mode which disables the cli access to the mobile's terminal).



```

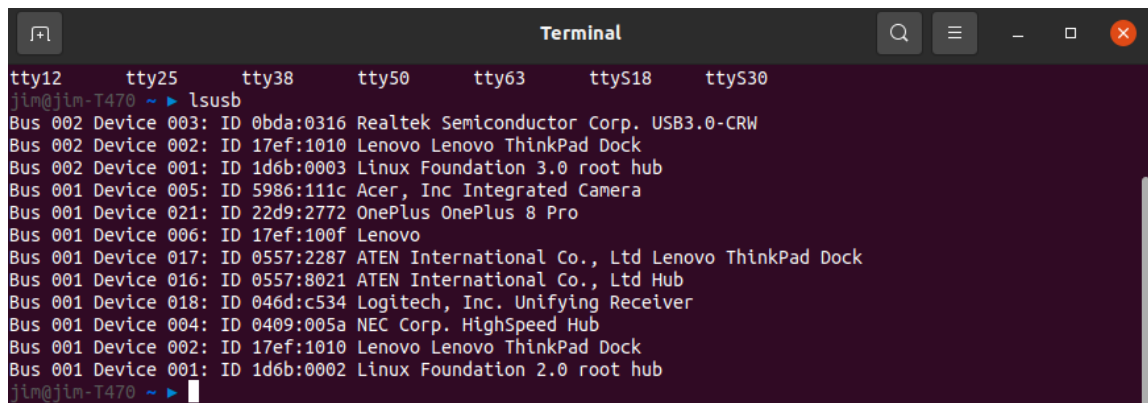
Terminal
jim@jim-T470 ~ ▶ adb devices
List of devices attached
c6759ac5      device

jim@jim-T470 ~ ▶ adb shell
OnePlus8Pro:/ $ su
OnePlus8Pro:/ # setprop sys.usb.config diag,adb
OnePlus8Pro:/ # jim@jim-T470 ~ ▶ █

```

Figure 20- ADB Diagnostic Mode

By running `lsusb` on the PC, a list containing devices with identifications and hexadecimal values referring to the vendor and product of each device will appear (Figure 22). Notice that the output of the above command changes from (Figure 21) (OnePlus 8 Pro 22d9:2772) to (Figure 22) (OnePlus 8 Pro 22d9:276f) the diagnostic mode. The information provided by the `lsusb` command is used for the next step, which is to inform the kernel of the OS where to expose the information deriving from the mobile's USB interface. So, by running `modprobe usbserial vendor=0x22d9 product=0x276f`, the messages from mobile's USB interface will be exposed on a serial device under `/dev/ttyUSBx` where 'x' could be any number from zero to depending on how many serial devices are connected to the PC.

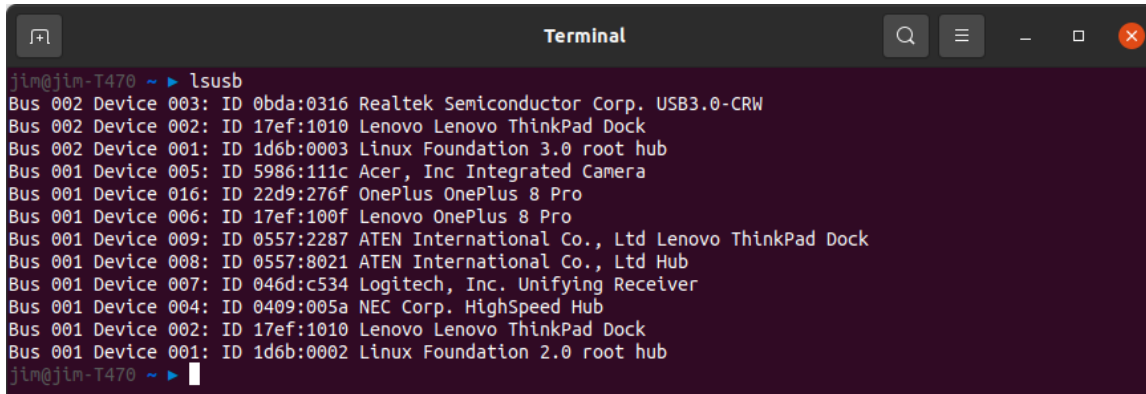


```

Terminal
tty12  tty25  tty38  tty50  tty63  ttyS18  ttyS30
jim@jim-T470 ~ ▶ lsusb
Bus 002 Device 003: ID 0bda:0316 Realtek Semiconductor Corp. USB3.0-CRW
Bus 002 Device 002: ID 17ef:1010 Lenovo Lenovo ThinkPad Dock
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 5986:111c Acer, Inc Integrated Camera
Bus 001 Device 021: ID 22d9:2772 OnePlus OnePlus 8 Pro
Bus 001 Device 006: ID 17ef:100f Lenovo
Bus 001 Device 017: ID 0557:2287 ATEN International Co., Ltd Lenovo ThinkPad Dock
Bus 001 Device 016: ID 0557:8021 ATEN International Co., Ltd Hub
Bus 001 Device 018: ID 046d:c534 Logitech, Inc. Unifying Receiver
Bus 001 Device 004: ID 0409:005a NEC Corp. HighSpeed Hub
Bus 001 Device 002: ID 17ef:1010 Lenovo Lenovo ThinkPad Dock
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
jim@jim-T470 ~ ▶ █

```

Figure 21- lsusb output before diagnostic mode



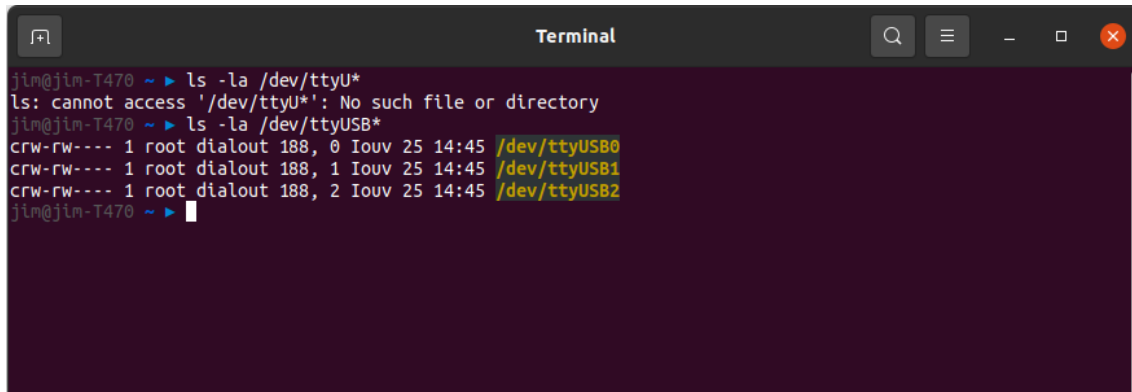
```

Terminal
jim@jim-T470 ~ ▶ lsusb
Bus 002 Device 003: ID 0bda:0316 Realtek Semiconductor Corp. USB3.0-CRW
Bus 002 Device 002: ID 17ef:1010 Lenovo Lenovo ThinkPad Dock
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 5986:111c Acer, Inc Integrated Camera
Bus 001 Device 016: ID 22d9:276f OnePlus OnePlus 8 Pro
Bus 001 Device 006: ID 17ef:100f Lenovo OnePlus 8 Pro
Bus 001 Device 009: ID 0557:2287 ATEN International Co., Ltd Lenovo ThinkPad Dock
Bus 001 Device 008: ID 0557:8021 ATEN International Co., Ltd Hub
Bus 001 Device 007: ID 046d:c534 Logitech, Inc. Unifying Receiver
Bus 001 Device 004: ID 0409:005a NEC Corp. HighSpeed Hub
Bus 001 Device 002: ID 17ef:1010 Lenovo Lenovo ThinkPad Dock
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
jim@jim-T470 ~ ▶

```

Figure 22- lsusb after diagnostic mode

After this preparation, the mobile's messages should be exposed on one of the devices placed under the `/dev/ttyUSBx` path (Figure 23). There is no exact way to discover which device will provide the information needed, so each of them should be tested.



```

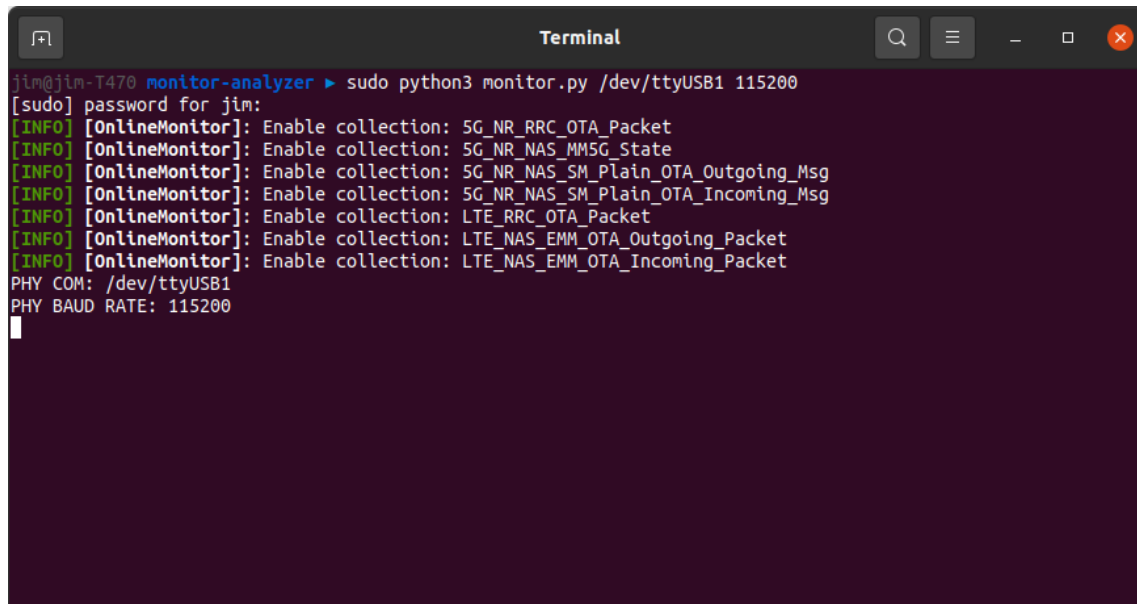
Terminal
jim@jim-T470 ~ ▶ ls -la /dev/ttyU*
ls: cannot access '/dev/ttyU*': No such file or directory
jim@jim-T470 ~ ▶ ls -la /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Iouv 25 14:45 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Iouv 25 14:45 /dev/ttyUSB1
crw-rw---- 1 root dialout 188, 2 Iouv 25 14:45 /dev/ttyUSB2
jim@jim-T470 ~ ▶

```

Figure 23- list ttyUSBx devices

MobileInsight software provides some examples located at `mobile-insight/mobileinsight-core/examples/`, which can be used for testing the functionality between the mobile and the MobileInsight software. One last piece of information is needed before running the MobileInsight python scripts, and it has to do with the baudrate of the usb port. Unfortunately, since diagnostic mode was not designed for public usage, usually there is no information regarding device's baudrate. The solution is to test known usb serial baudrate values and find which is suitable. Baudrate values range from a list including the values 110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 128000 and 256000. As technology advances, the chances with today's mobile phones are on higher baudrates. For the OnePlus 8 Pro which is used in this experiment, the baudrate was found to be 115200. Two custom scripts were used; `monitor.py`, and `offline-analysis.py` for monitoring and analysing the captured messages accordingly. Moreover, for testing the MobileInsight application

and the captured messages, a private 5G network (AMARI Callbox Mini) was used. The mobile was in airplane mode for not transmitting any messages, and prepared the MobileInsight scripts (Figure 24).

A terminal window titled "Terminal" with a dark background. The prompt is "jim@jim-T470 monitor-analyzer". The command entered is "sudo python3 monitor.py /dev/ttyUSB1 115200". The output shows several lines of green text: "[INFO] [OnlineMonitor]: Enable collection: 5G_NR_RRC_OTA_Packet", "[INFO] [OnlineMonitor]: Enable collection: 5G_NR_NAS_MMSG_State", "[INFO] [OnlineMonitor]: Enable collection: 5G_NR_NAS_SM_Plain_OTA_Outgoing_Msg", "[INFO] [OnlineMonitor]: Enable collection: 5G_NR_NAS_SM_Plain_OTA_Incoming_Msg", "[INFO] [OnlineMonitor]: Enable collection: LTE_RRC_OTA_Packet", "[INFO] [OnlineMonitor]: Enable collection: LTE_NAS_EMM_OTA_Outgoing_Packet", and "[INFO] [OnlineMonitor]: Enable collection: LTE_NAS_EMM_OTA_Incoming_Packet". Below this, it shows "PHY COM: /dev/ttyUSB1" and "PHY BAUD RATE: 115200".

```
jim@jim-T470 monitor-analyzer ▶ sudo python3 monitor.py /dev/ttyUSB1 115200
[sudo] password for jim:
[INFO] [OnlineMonitor]: Enable collection: 5G_NR_RRC_OTA_Packet
[INFO] [OnlineMonitor]: Enable collection: 5G_NR_NAS_MMSG_State
[INFO] [OnlineMonitor]: Enable collection: 5G_NR_NAS_SM_Plain_OTA_Outgoing_Msg
[INFO] [OnlineMonitor]: Enable collection: 5G_NR_NAS_SM_Plain_OTA_Incoming_Msg
[INFO] [OnlineMonitor]: Enable collection: LTE_RRC_OTA_Packet
[INFO] [OnlineMonitor]: Enable collection: LTE_NAS_EMM_OTA_Outgoing_Packet
[INFO] [OnlineMonitor]: Enable collection: LTE_NAS_EMM_OTA_Incoming_Packet
PHY COM: /dev/ttyUSB1
PHY BAUD RATE: 115200
```

Figure 24- MobileInsight monitor before capture

Afterwards, airplane mode was disabled, and the capture started (Figure 25). The capture can be stopped by pressing `ctrl+c` (SIGINT) on the terminal. Then, a `mi2log` file will be created in the same path with the script that was executed, the created file contains all the messages that was captured. Decoded information is displayed in the terminal in readable format and can also be retrieved from the `mi2log` file by running the `offline-analysis.py` script (Figure 26). The `offline-analysis` script creates a file in readable format (Figure 27). The part with the `offline-analysis` can be skipped, user can use preferably the `mi-gui`. After running the `mi-gui` command and the window appears, the user can open the `mi2log` file he captured in the previous step and scroll through the captured messages in a more user-friendly environment (Figure 28). User can select the desired message, it then expands in the right side of the window, and from there, the user can expand or compress the fields of interest.


```

Terminal
jim@jim-T470 monitor-analyzer ▶ sudo python3 monitor.py /dev/ttyUSB1 115200
[sudo] password for jim:
[INFO] [OnlineMonitor]: Enable collection: 5G_NR_RRC_OTA_Packet
[INFO] [OnlineMonitor]: Enable collection: 5G_NR_NAS_MM5G_State
[INFO] [OnlineMonitor]: Enable collection: 5G_NR_NAS_SM_Plain_OTA_Outgoing_Msg
[INFO] [OnlineMonitor]: Enable collection: 5G_NR_NAS_SM_Plain_OTA_Incoming_Msg
[INFO] [OnlineMonitor]: Enable collection: LTE_RRC_OTA_Packet
[INFO] [OnlineMonitor]: Enable collection: LTE_NAS_EMM_OTA_Outgoing_Packet
[INFO] [OnlineMonitor]: Enable collection: LTE_NAS_EMM_OTA_Incoming_Packet
PHY COM: /dev/ttyUSB1
PHY BAUD RATE: 115200
[INFO] [MsgLogger]: <dm_log_packet><pair key="log_msg_len">38</pair><pair key="type_id">5G_NR_NAS_MM5G_Stat
e</pair><pair key="timestamp">2022-06-22 13:04:14.6743783</pair><pair key="Version">1</pair><pair key="MM5G
State">DEREGISTERED</pair><pair key="Mm5g Deregistered Substate">1</pair><pair key="PLMN Id">{0x0,0x0,0x0}<
/pair><pair key="GUTI" type="dict"><dict><pair key="UE Id">0x2</pair><pair key="GUTI PLMN ID">{0x0,0xf1,0x1
0}</pair><pair key="AMF Region ID">0x80</pair><pair key="AMF Set ID">0x0400</pair><pair key="AMF Pointer">0
x01</pair><pair key="5G TMSI">{0xb8,0x39,0x8a,0x91}</pair></dict></pair><pair key="MM5G Update Status">UPDA
TED</pair><pair key="Octet[0]">0</pair><pair key="Octet[1]">0</pair><pair key="Octet[2]">0</pair></dm_log_p
acket>
(MI)Unknown 5G NR RRC OTA Message version: 0xc
[INFO] [MsgLogger]: <dm_log_packet><pair key="log_msg_len">39</pair><pair key="type_id">5G_NR_RRC_OTA_Packe
t</pair><pair key="timestamp">2022-06-22 13:04:14.640232</pair><pair key="Pkt Version">12</pair><pair key="
Unknown">0</pair><pair key="RRC Release Number">15</pair><pair key="RRC Version Number">0x90</pair><pair ke
y="Radio Bearer ID">255</pair><pair key="Physical Cell ID">500</pair><pair key="Freq">631968</pair><pair ke
y="5G FreqNum">5</pair><pair key="5G FreqNum">16777316</pair><pair key="RPH Number">0</pair><pair key="CTP
Back to 5G">0</pair>

```

Figure 25- MobileInsight captured messages

For validating and recognizing the captured files and the scripts created for this experiment, a crosscheck was made between the Amarisoft Callbox Mini, and MobileInsight software.

```

Terminal
jim@jim-T470 monitor-analyzer ▶ python3 offline-analysis.py monitor.mt2log
[INFO] [OfflineReplayer]: Enable LTE_PHY_Serv_Cell_Measurement
[INFO] [OfflineReplayer]: Enable LTE_NB1_ML1_GM_DCI_Info
[INFO] [OfflineReplayer]: Enable 5G_NR_RRC_OTA_Packet
[INFO] [OfflineReplayer]: Enable 5G_NR_NAS_MM5G_State
[INFO] [OfflineReplayer]: Enable 5G_NR_NAS_SM_Plain_OTA_Outgoing_Msg
[INFO] [OfflineReplayer]: Enable 5G_NR_NAS_SM_Plain_OTA_Incoming_Msg
[INFO] [OfflineReplayer]: Enable LTE_RRC_OTA_Packet
[INFO] [OfflineReplayer]: Enable LTE_NAS_EMM_OTA_Outgoing_Packet
[INFO] [OfflineReplayer]: Enable LTE_NAS_EMM_OTA_Incoming_Packet
Init NR RRC Analyzer
Init RRC Analyzer
[INFO] [OfflineReplayer]: Enable LTE_RRC_Serv_Cell_Info
[INFO] [OfflineReplayer]: Enable LTE_RRC_CDRX_Events_Info
[INFO] [OfflineReplayer]: Enable WCDMA_RRC_OTA_Packet
[INFO] [OfflineReplayer]: Enable WCDMA_RRC_Serv_Cell_Info
[INFO] [OfflineReplayer]: Enable WCDMA_RRC_States
[INFO] [OfflineReplayer]: Enable LTE_NAS_ESM_OTA_Incoming_Packet
[INFO] [OfflineReplayer]: Enable LTE_NAS_ESM_OTA_Outgoing_Packet
[INFO] [OfflineReplayer]: Enable LTE_NAS_EMM_State
[INFO] [OfflineReplayer]: Enable LTE_NAS_ESM_State
[INFO] [UmtsNasAnalyzer]: Initialing UmtsNasAnalyzer..
[INFO] [OfflineReplayer]: Enable UMTS_NAS_OTA_Packet
[INFO] [OfflineReplayer]: Enable UMTS_NAS_GMM_State

```

Figure 26- MobileInsight offline analysis

Among other captured messages as point of reference 5G-TMSI can be used. 5G-TMSI as already mentioned before is a temporary identifier used for identifying the UE on the AMF. Its value changes periodically, and it is mapped to the IMSI or SUPI (depends on vendor's configuration). In Figure 27 and Figure 28 is highlighted the captured 5G-TMSI with value 0x918a39b8, and from Amarisoft's web-gui we can verify this information (Figure 29).

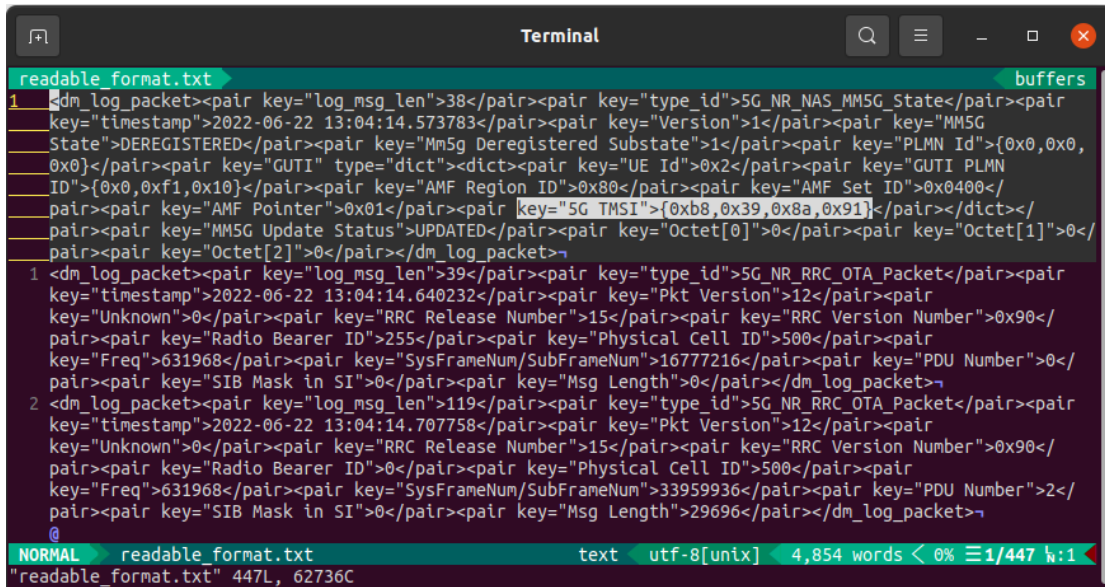


Figure 27- output file of offline-analysis.py

In Figure 29 notice that the service running is 5G in SA mode, however Amarisoft have not changed yet some fields from 4G to 5G, i.e., MME instead of AMF, ENB instead of GNB and on the up left corner of the image it LTE instead of 5G.

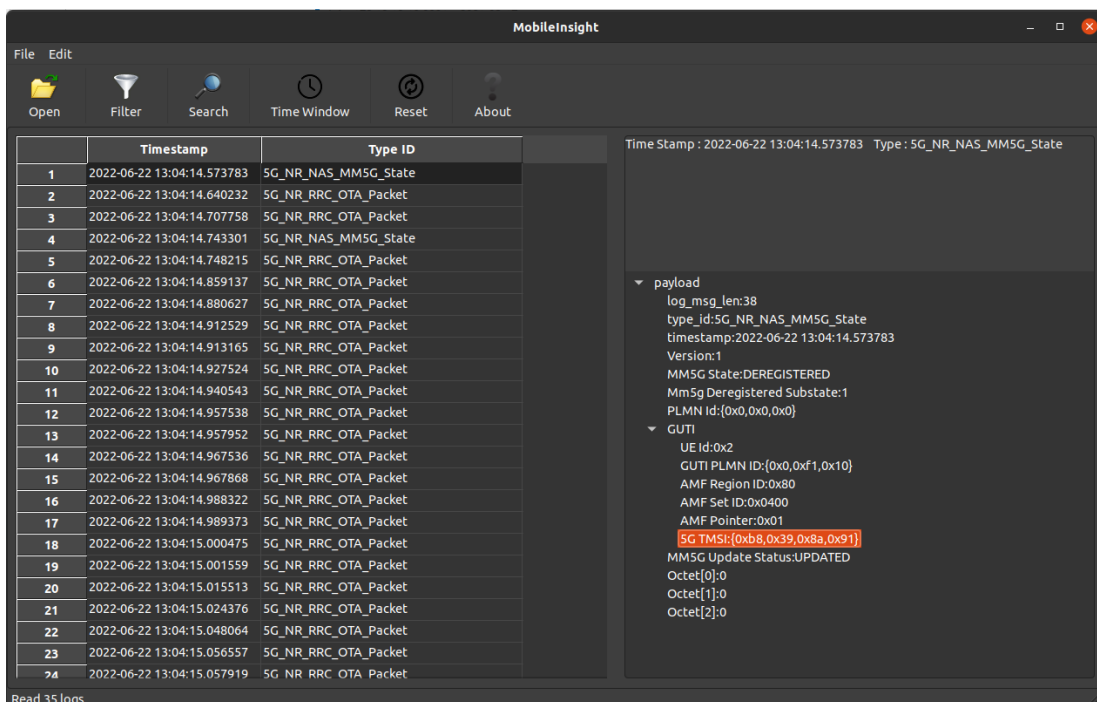


Figure 28- MobileInsight GUI (mi-gui)

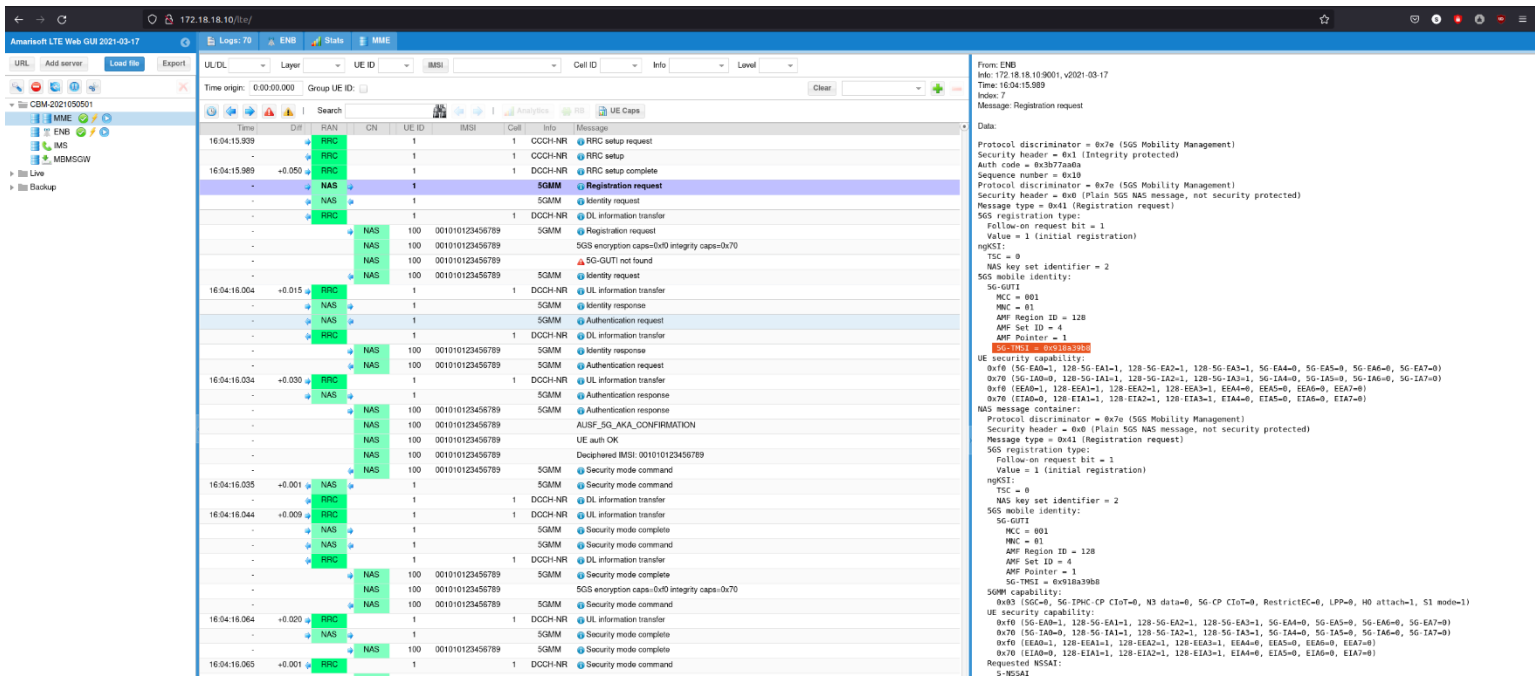


Figure 29- Amarisoft GUI 5G-TMSI

Evaluation of Captured Messages

As mentioned before the experiments were performed on a commercial mobile phone (OnePlus 8 Pro) having root access and diagnostic mode enabled, connected with a laptop equipped with MobileInsight software. Vodafone and Cosmote are two major vendors providing 5G mobile services, and they were selected for evaluation. Intrigued from [22], the evaluation is focused on messages regarding; the cryptographic algorithms that MNOs are using; the frequency of AKA between UE and MME (deliberately used MME instead of AMF, as current 5G deployments are in NSA mode, depending on 4G Core); the frequency of TMSI reallocations; the timer that TMSI is kept while the UE is disconnected (before the MME requests for the IMSI).

Results

Regarding supported cipher and integrity algorithms the phone was disconnected from the network (airplane mode enabled), MobileInsight software was in monitor mode (running on the Desktop environment) ready to capture RRC and NAS messages. Then the airplane mode was disabled on the UE, and while the UE was trying to connect to the gNB, the negotiation between them was captured. The same experiment was done for both operators by swapping USIM cards. (Captured files and further guidance are uploaded in GitHub¹). In Table 1, the only difference between the two operators is that Vodafone supports GEA4 (in contrast with Cosmote) and has excluded support for GEA1 and GEA2 which are considered insecure (where Cosmote still supports them).

NAME	Integrity	Cipher	Vodafone	Cosmote
SNOW 3G	128-EIA1	128-EEA1	✓	✓
AES	128-EIA2	128-EEA2	✓	✓
ZUC	128-EIA3	128-EEA3	✓	✓
KASUMI	UIA1	UEA1	✓	✓
-	128-5G-IA1	128-5G-EA1	✓	✓
-	128-5G-IA2	128-5G-EA2	✓	✓
-	128-5G-IA3	128-5G-EA3	✓	✓
No encryption designation for LTE		EEA0	✓	✓
No encryption designation for 3G		UEA0	✓	✓
		5G-EA0	✓	✓
GPRS encryption algorithm 1		GEA1		✓
GPRS encryption algorithm 2		GEA2		✓
GPRS encryption algorithm 3		GEA3	✓	✓
GPRS encryption algorithm 4		GEA4	✓	

Table 1- Ciphering and integrity algorithms used from Vodafone & Cosmote

SNOW3G and KASUMI were used in 3G, and they were made public. The shared key that is used for encryption is 128 bits long. This key length is considered sufficient. The 4G (and current 5G) encryption algorithms are based on SNOW3G, AES and ZUC. All encryption functions in all generations of mobile communications technology are stream ciphers. Both AES and KASUMI are block ciphers, but they are used in stream

¹ https://github.com/santojim/minsight_captured_files/tree/main/ciphering_and_integrity_algs

cipher mode. Stream cipher is suitable for high-speed communication scenarios because a major part of encryption/decryption can be done already before the plaintext/ciphertext is even available. GEA1 and GEA2 are rather insecure encryption algorithms covering the entire GSM/GPRS/EDGE/3G/4G spectrum [23], which can be decrypted. GEA/3 is based on the 64-bit version of KASUMI, while the GEA/4 offers the advanced 128-bit version. Both cover the full cellular spectrum GSM/GPRS/EDGE/3G/4G. All GEA-type encryption algorithms focus on providing encryption for all other non-voice data packets.

Regarding the AKA the mobile was connected to the network in idle mode, meaning no calls, no messages, and no airplane mode triggered. The MobileInsight was running for 60 minutes for the case of Vodafone and 70 minutes for the case of Cosmote (captured files with timestamps are available at GitHub²). Using the mi-gui software and opening the files from frequency_of_aka_keys_change directory, then using the search button and searching for the word RAND, the results in Figure 30 appear. The assumption that the AKA was triggered is perceived from the fact that RAND was being sent to the UE together with authentication values containing SQN, AMF, and MAC values.

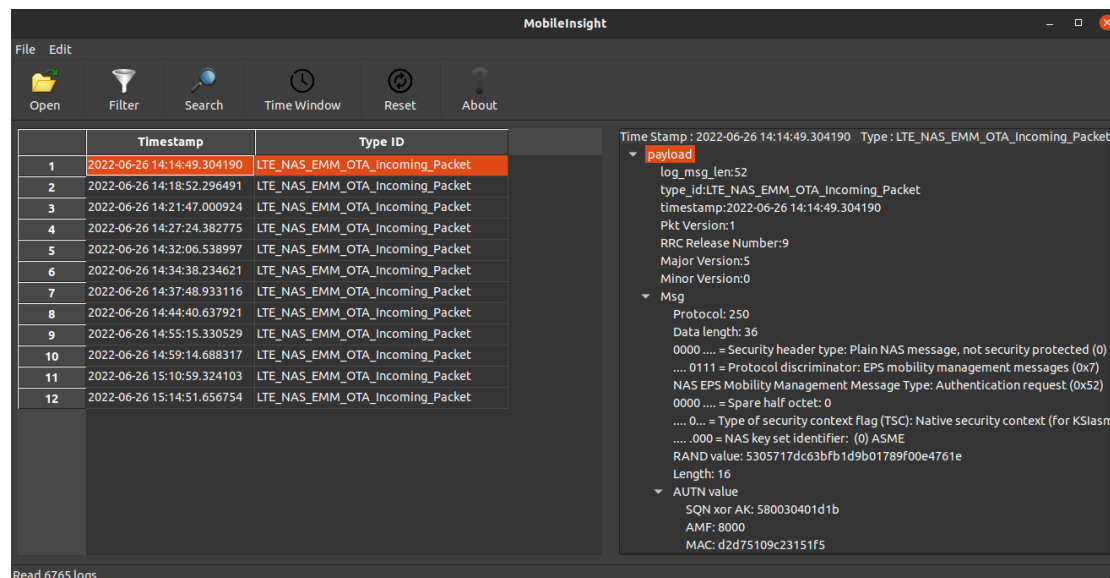


Figure 30- mi-gui search RAND

² https://github.com/santojimi/minsight_captured_files/tree/main/frequency_of_aka_keys_change

In Table 2 in the left column is a counter of the key refreshment attempt, in the middle column there is the minutes between two keys refreshments for Vodafone and in the right column the same for Cosmote. The timer started counting from the first key refreshment until the last one. As the time between AKAs is not fixed, and no pattern can be perceived, the average number of keys exchanged for each vendor will be measured in the corresponding time frame. In the last row there is the average, for Vodafone it is measured as one key refreshment per 5.45 minutes and for Cosmote a slightly longer time of one key refreshment per 7 minutes.

Key refresh	Vodafone (in minutes)	Cosmote (in minutes)
1	4	5
2	3	8
3	6	9
4	5	6
5	2	7
6	3	8
7	9	4
8	9	6
9	4	9
10	9	8
11	4	-
Avg	1 key change every 5 minutes and 27 seconds	1 key change every 7 minutes

Table 2 - AKA frequency

Regarding the TMSI reallocation, an issue regarding security was noticed on both vendors. While TMSIs changed for both vendors by triggering airplane mode, the new assigned values were not unpredictable as [24] mentions. In this experiment, airplane mode was triggered four times for Cosmote and 5 times for Vodafone in a short period of time as can be verified from captured files³. TMSIs should change frequently and use random, unpredictable values, otherwise as it has been proved from [25] and [26], an attacker can locate if a subscriber's device is present in the same area where the attack is taking place. In Table 3 and Table 4 hex values with red color represent the values that have not changed for the new allocated TMSI and by green color the values that changed.

³ https://github.com/santojim/minisight_captured_files/tree/main/TMSI_change_frequency

Timestamp	TMSIs Cosmote
2022-06-11 14:42:59.740609	0xc74cd98f
2022-06-11 14:45:02.438784	0xc74dd912
2022-06-11 14:45:31.552092	0xc74dd930
2022-06-11 14:45:56.704130	0xc74dd93f
2022-06-11 14:46:22.596962	0xc74dd982

Table 3 - TMSIs captured from Cosmote

Timestamp	TMSIs Vodafone
2022-06-11 14:52:18.745387	0xd4fab7cc
2022-06-11 14:53:38.594135	0xd4fab7d4
2022-06-11 14:53:38.594135	0xd4fab7e5
2022-06-11 14:54:26.732135	0xd4fab7f2
2022-06-11 14:55:32.196671	0xd4fab7fd

Table 4- TMSIs captured from Vodafone

A better application of TMSI reallocation is demonstrated in Table 5 from the local 5G-SA network by Amarisoft while trying the same experiment, triggering airplane mode on and off. For the values presented in Table 5 the mi-gui software was used, and the file opened was `amarisoft_airplane_on_off_4_times.mi2log`⁴. By searching for `REGISTERED_INITIATED`, the results below will be listed.

Timestamp	TMSIs Amarisoft
2022-06-11 11:56:11.751993	0x8866b448
2022-06-11 11:56:30.151900	0xb682f6b3
2022-06-11 11:56:36.871981	0x9b1df5f9
2022-06-11 11:56:44.871851	0xc325d658
2022-06-11 11:57:08.871824	0x25xe281e

Table 5- TMSIs captured from Amarisoft

For further investigating the vendor's conformity regarding the 3GPP technical specifications and their implementations regarding the TMSI reallocations, the mobile was monitored for about eight hours connected to the mobile network with data enabled, in idle state. The captured files are available for both vendors in files `vodafone_9_hours_still.mi2log` and `cosmote_8_hours_still.mi2log`⁴. The results are quite interesting, Vodafone made a TMSI reallocation after 5 hours and the TMSI changed from `0xdce9b77e` to `0xdd40b771`. Marked with red are the hexadecimal values that did not change. The new allocated TMSI can be considered sufficient compared to

⁴ https://github.com/santojimi/minisight_captured_files/tree/main/TMSI_change_frequency

when triggering airplane on and off, however from the eight hex values only four changed, which could lead to correlation with the old TMSI. After this change, in a time frame of an hour the MME requested the IMSI twenty-five times. This means that either an attack with a false base station took place at that time, or Vodafone has abused the capability of requesting the IMSI. These results (Figure 31) can be verified by opening the Vodafone mi2log, using the mi-gui software and searching for “TMSI:”.

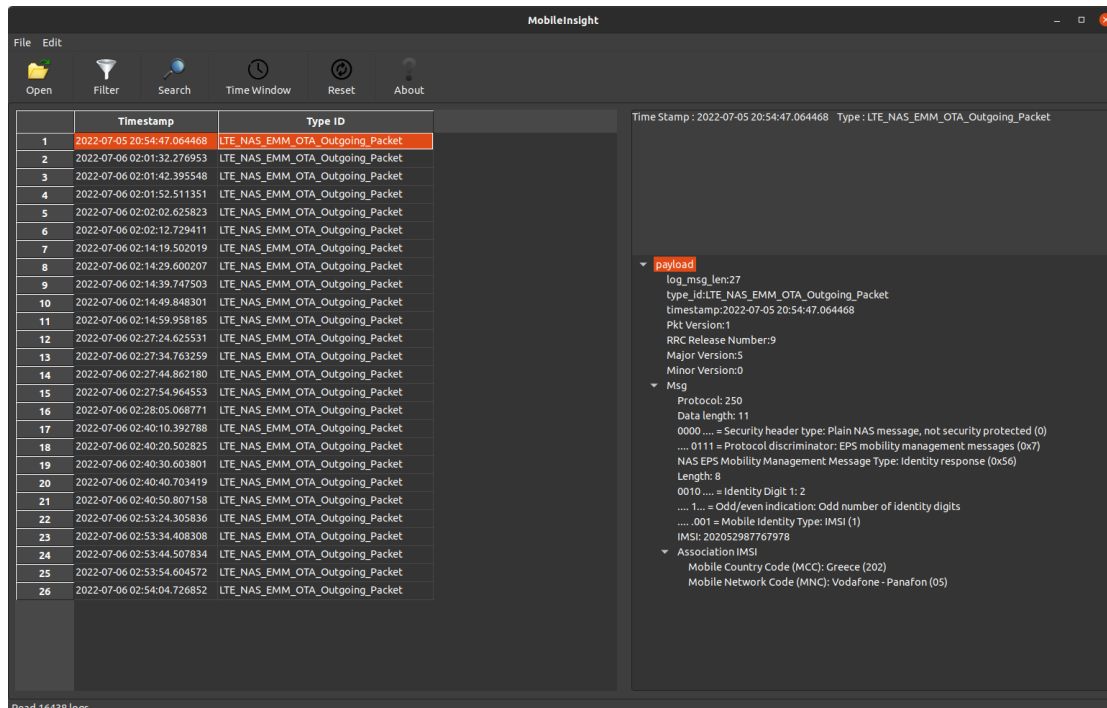


Figure 31- Vodafone IMSI sent 26 times

For Cosmote, in a time frame of almost 8 hours, even though TMSI did not change, IMSI was not requested even once. Another notice is that the captured mi2log file from Cosmote was four times larger than the one from Vodafone (909kb for Vodafone versus 4,19Mb for Cosmote), which made the mi-gui crash many times while trying to open it.

The above results triggered another question regarding the time that the vendors keep alive a TMSI even when a UE is offline. So, by using airplane mode on and off for different time slots, the mobile was left offline for 5 then 10, 15, 20, 25 and 30 minutes respectively. When the airplane mode was off (while the UE tried to connect to the mobile network), the TMSI was tracked from the log files, and checked if it was accepted from the MME. If the MME did not recognize the TMSI, it would request the

IMSI. For Vodafone, the time recognizing the TMSI was 25 minutes, however even that MME recognized the TMSI, it issued the UE a GUTI reallocation command, which caused the TMSI to change. In 30 minutes, it requested the IMSI. The exact same method was used for Cosmote, and the results were identical. The captured files can be found on GitHub⁵.

Summary

Apropos of AKA execution frequency, given that UE was idle (no calls, no SMS, not any application requesting data) in both cases of Vodafone and Cosmote, the keys were updated almost ten times in a frame of one hour which is not ideal, considering that AKA should be performed after each call, SMS, or data session, which unfortunately was not the case in the experiments. Furthermore, regarding the cipher algorithms Vodafone has a slight lead over Cosmote for not supporting GEA1 and GEA2 insecure encryption algorithms. Results regarding user privacy are disappointing considering the TMSI reallocations that were monitored for both vendors. A slightly better performance will be considered the one from Cosmote as it did not request the IMSI, in contrast with Vodafone which in a frame of 9 hours the IMSI was requested 25 times. However, both vendors lack procedures of TMSI reallocations during calls, SMS exchanges, data usage, which combined with the poorly random reallocated TMSIs (soft power on/off), can expose subscriber's location [27]. These issues exist, due to the selected 5G deployment, which is based on 4G Core network and therefore inherits all the vulnerabilities. In a mature 5G SA deployment these issues could be resolved as it was observed in the experiment with the local Amarisoft 5G network. Moreover, the use of SUCI instead of plaintext IMSI for the initial reconnaissance of the subscriber in the home mobile network will diminish the current privacy exposure.

⁵ https://github.com/santojim/minsight_captured_files/tree/main/IMSI_sent

Conclusion

Mobile network operators have invested a considerable amount of effort into implementing guidelines defined by national standards development organizations, targeting the protection of their subscriber's privacy. Unfortunately, subscribers are still exposed to several attacks, some of which are even inherited from previous radio generations. Moreover, as years pass and radio generations advances, investments are made for the coverage of vast geographical areas. However, this means that network operators need to struggle to update their equipment in all these territories. In addition, they need to support previous radio generations, as they cannot force their subscribers to update their equipment (mobile phones). Therefore, vulnerabilities found on older radio generations (i.e., 2G, 3G) are still effective by downgrade attacks, where false base stations emit stronger signals and use the same network identifier as the original, deluding the subscribers to connect to the rogue base station. There is not a direct solution to the problems mentioned above, time is needed for both mobile network operators, and the subscribers. Apart from the physical issues mentioned above, in the 5G era a new issue has arisen for some countries that are not fond of the privacy that 5G is offering for the end user, and this is probably the reason that some of the standards developed for 5G are flexible, and up to service providers to implement them.

Scripts

Custom Plugin for MobileInsight Desktop Interface

Filename: monitor.py

```
#!/usr/bin/python
# Filename: monitor.py
import os
import sys

# Import MobileInsight modules
from mobile_insight.monitor import OnlineMonitor
from mobile_insight.analyzer import MsgLogger

if __name__ == "__main__":

    if len(sys.argv) < 3:
        print("Error: please specify physical port name and baudrate.")
        print((__file__, "SERIAL_PORT_NAME BAUDRATE"))
        sys.exit(1)

    # Initialize a 3G/4G monitor
    src = OnlineMonitor()
    src.set_serial_port(sys.argv[1]) # the serial port to collect the
traces
    src.set_baudrate(int(sys.argv[2])) # the baudrate of the port

    # Save the monitoring results as an offline log
    src.save_log_as("./monitor.mi2log")

    # Enable 4G/5G messages to be monitored. Here we enable RRC (radio
# resource control) monitoring
    src.enable_log("5G_NR_RRC_OTA_Packet")
    src.enable_log("5G_NR_NAS_MM5G_State")
    src.enable_log("5G_NR_NAS_SM_Plain_OTA_Outgoing_Msg")
    src.enable_log("5G_NR_NAS_SM_Plain_OTA_Incoming_Msg")
    src.enable_log("LTE_RRC_OTA_Packet")
    src.enable_log("LTE_NAS_EMM_OTA_Outgoing_Packet")
    src.enable_log("LTE_NAS_EMM_OTA_Incoming_Packet")

    # Dump the messages to std I/O.
    dumper = MsgLogger()
    dumper.set_source(src)
    dumper.set_decoding(MsgLogger.XML) # decode the message as xml

    # Start the monitoring
    src.run()
```

Filename: offline-analysis.py

```
#!/usr/bin/python
# Filename: offline-analysis.py
import os
import sys

"""
Offline analysis by replaying logs
"""

# Import MobileInsight modules
from mobile_insight.monitor import OfflineReplayer
from mobile_insight.analyzer import MsgLogger, NrRrcAnalyzer,
LteRrcAnalyzer, WcdmaRrcAnalyzer, LteNasAnalyzer, UmtsNasAnalyzer,
LteMacAnalyzer, LteMeasurementAnalyzer

if __name__ == "__main__":

    # Initialize a monitor
    src = OfflineReplayer()
    src.set_input_path("monitor.mi2log")

    src.enable_log("5G_NR_RRC_OTA_Packet")
    src.enable_log("5G_NR_NAS_MM5G_State")
    src.enable_log("5G_NR_NAS_SM_Plain_OTA_Outgoing_Msg")
    src.enable_log("5G_NR_NAS_SM_Plain_OTA_Incoming_Msg")
    src.enable_log("LTE_RRC_OTA_Packet")
    src.enable_log("LTE_NAS_EMM_OTA_Outgoing_Packet")
    src.enable_log("LTE_NAS_EMM_OTA_Incoming_Packet")
    src.enable_log("LTE_PHY_Serv_Cell_Measurement")
    src.enable_log("LTE_NB1_ML1_GM_DCI_Info")

    logger = MsgLogger()
    logger.set_decode_format(MsgLogger.XML)
    logger.set_dump_type(MsgLogger.FILE_ONLY)
    logger.save_decoded_msg_as("./readable_format.txt")
    logger.set_source(src)

    # Analyzers
    nr_rrc_analyzer = NrRrcAnalyzer()
    nr_rrc_analyzer.set_source(src) # bind with the monitor

    lte_rrc_analyzer = LteRrcAnalyzer()
    lte_rrc_analyzer.set_source(src) # bind with the monitor

    wcdma_rrc_analyzer = WcdmaRrcAnalyzer()
    wcdma_rrc_analyzer.set_source(src) # bind with the monitor

    lte_nas_analyzer = LteNasAnalyzer()
    lte_nas_analyzer.set_source(src)

    umts_nas_analyzer = UmtsNasAnalyzer()
    umts_nas_analyzer.set_source(src)

    lte_mac_analyzer = LteMacAnalyzer()
    lte_mac_analyzer.set_source(src)

    lte_meas_analyzer = LteMeasurementAnalyzer()
```

```
lte_meas_analyzer.set_source(src)

# Start the monitoring
src.run()
```

Custom Plugin for MobileInsight Mobile Interface

main.mi2app

```
# main.mi2app
import os
import sys

from mobile_insight.monitor import OnlineMonitor
from mobile_insight.analyzer import MsgLogger

# A helper utilities for mobileinsight
from service import mi2app_utils

# options of enabling logs

nr_control = [
    "5G_NR_RRC_OTA_Packet",
    "5G_NR_NAS_MM5G_State",
    "5G_NR_NAS_SM_Plain_OTA_Outgoing_Msg",
    "5G_NR_NAS_SM_Plain_OTA_Incoming_Msg",
    "LTE_RRC_OTA_Packet",
    "LTE_NAS_EMM_OTA_Outgoing_Packet",
    "LTE_NAS_EMM_OTA_Incoming_Packet",
]

# Initialize a 5G monitor
src = OnlineMonitor()

# Configure the path of saving logs
cache_directory = mi2app_utils.get_cache_dir()
log_directory = os.path.join(cache_directory, "mi2log")
src.set_log_directory(log_directory)

# Enable logs based on settings
if plugin_config["log_type"] == "5G Control Plane":
    src.enable_log(nr_control)

#Start the monitoring
src.run()
```

readme.txt

```
A simple analyzer for 5G only protocols
```

settings.json

```
[
  {
    "type" : "title",
    "title" : "5G_NAS_LOGS"
  },
  {
    "type" : "options",
    "title" : "Log Type",
    "desc" : "Set the log type to collect",
    "key" : "log_type",
    "default": "5G Control Plane",
    "options": ["5G Control Plane", "5G Control/Data Plane"]
  },
  {
    "type" : "numeric",
    "title" : "Log Size (KB)",
    "desc" : "The size of each log",
    "key" : "mi_log_size",
    "default": "1500"
  }
]
```

References

- [1] Padmanabhan and Arvind, "5G UE Data Rate," 25 04 2022. [Online]. Available: <https://devopedia.org/5g-ue-data-rate>.
- [2] "3gpp The Mobile Broadband Standard: About 3GPP," [Online]. Available: <https://www.3gpp.org/about-3gpp>. [Accessed 02 06 2022].
- [3] O. O. Erunkulu, A. M. Zungeru, C. K. Lebekwe, M. Mosalaosi and J. M. Chuma, "5G Mobile Communication Applications: A Survey and Comparison of Use Cases," *IEEE*, no. 14.06.2021, p. 45, 2021.
- [4] "LTE Network Architecture," [Online]. Available: tutorialspoint.com/lte/lte_network_architecture.htm. [Accessed 28 05 2022].
- [5] H. S. Rafiul, E. Mitziu, K. Imtiaz, C. Omar and B. Elisa, "5GReasoner: A Property-Directed Security and Privacy Analysis Framework for 5G Cellular Network Protocol," *CCS*, p. 15, 2019.
- [6] D. Koziol and H.-L. Määtänen, "Network Architecture and NR Radio Protocols," in *5G New Radio: A Beam-based Air Interface, First Edition.*, John Wiley & Sons Ltd, 2020.
- [7] "5G NR Radio Protocol Stack," 4 9 2017. [Online]. Available: <https://www.techplayon.com/5g-nr-radio-protocol-stack-layer-2-layer-3/>. [Accessed 2022 06 08].
- [8] "5G Standalone Access Registration Signaling Messages," [Online]. Available: <https://www.eventhelix.com/5g/standalone-access-registration/details/5g-standalone-access-registration.html#preamble>. [Accessed 03 06 2022].
- [9] 3GPP, "Procedures for the 5G System (TS 23.502 v17.4.0)," 2022.
- [10] ETSI, "5G Security architecture and procedures for 5G System 3GPP TS 33.501 version 15.2.0 release 15," ETSI, Sophia Antipolis, 2018.

- [11] J. Arkko, V. Lehtovirta and P. Eronen, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)," Internet Engineering Task Force (IETF).
- [12] 3GPP, "Security architecture and procedures for 5G system (TS 33.501 v17.5.0 2022-03)," 2022.
- [13] D. Simon, B. Aboba and R. Hurst, "The EAP-TLS Authentication Protocol," 2008.
- [14] J. Mattsson and P. K. Nakarmig, "Nori: Concealing the Concealed Identity, Availability, Reliability and Security (ARES 2021), Vienna, Austria, 2021.
- [15] 3GPP, "System Architecture for the 5G System (TS 23.501 v17.4.0 2022-03)," 2022.
- [16] D. Bubley, "Monetizing 5G NSA: do MNOs need to wait for standalone cores?," 27 07 2021. [Online]. Available: <https://www.amdocs.com/insights/blog/monetizing-5g-nsa-do-mnos-need-wait-standalone-cores>. [Accessed 2022].
- [17] Y. Li, C. Peng, Z. Zhang, Z. Tan, H. Deng, J. Zhao, Q. Li, Y. Guo, K. Ling, B. Ding, H. Li and S. Lu, "Experience: A Five-Year Retrospective of MobileInsight," ACM MobiCom, New Orleans, LA, USA, 2021.
- [18] "QCSuper," [Online]. Available: <https://github.com/P1sec/QCSuper>. [Accessed 2022].
- [19] Y. Li, C. Peng, Z. Zhang, Z. Tan, H. Deng, J. Zhao, Q. Li, Y. Guo, K. Ling, B. Ding, H. Li and S. Lu, "MobileInsight," [Online]. Available: <https://github.com/mobile-insight/mobileinsight-core>. [Accessed 2022].
- [20] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng and T. Wang, "MobileInsight: Extracting and Analyzing Cellular Network," MobiCom, NY, USA, 2016.
- [21] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng and T. Wang, "MobileInsight," [Online]. Available: <http://mobileinsight.net/>. [Accessed 2022].

- [22] C. Xenakis, C. Ntantogian and O. Panos, "(U)SimMonitor: A Mobile Application for Security Evaluation of Cellular Networks," 2016.
- [23] The GSM Association, "Security Algorithm Implementation Roadmap Official Document FS.35," 2020.
- [24] 3GPP, "Security Architecture TS 33.102 v17.0.0.0," 3GPP, 2022.
- [25] S. R. Hussain, M. Echeverriay, O. Chowdhuryy, N. Li and E. Bertino, "Privacy Attacks to the 4G and 5G Cellular Paging Protocols Using Side Channel Information," *Network and Distributed Systems Security (NDSS)*, 2019.
- [26] M. Arapinis, L. I. Mancini, E. Ritter and M. Ryan, "Privacy through Pseudonymity in Mobile Telephony Systems," *Network and Distributed Systems Security (NDSS) Symposium*, 2014.
- [27] A. Shaik, R. Borgaonkary, N. Asokanz, V. Niemix and J.-P. Seifert, "Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems," *NDSS*, 2017.