UNIVERSITY OF PIRAEUS

DEPARTMENT OF DIGITAL SYSTEMS

Msc DIGITAL SYSTEMS SECURITY

Master Thesis

Security Assessment of GSM Um Interface using SDR Systems

Papadopoulos P. Loukas MTE1823

Supervisor

Dr. Christos Xenakis

Piraeus, April 2021

Abstract

Lately, we have been experiencing an increasing popularity of LTE and 5G technologies. This popularity, however, doesn't seem to be enough to limit the deployment and usage of the oldest GSM cellular networks. The fact is that the GSM technology is active and will remain active and supported by most telecommunications providers for many years to come. According to Informa, there is an estimated 4.6 billion GSM subscribers worldwide, and this raises the key question of whether a ~20-year-old technology that serves such a large percentage of devices is secure. Another worrying question is how easily an attacker can cause downgrade to GSM technology, from the seemingly invulnerable LTE networks. In this master thesis we will try to answer these questions both theoretically and practically as well as to prove that as long as the GSM network infrastructure is active, the security provided by the new LTE networks can be pbypassed. In other words, we will prove the infallible law of security, which also applies to mobile networks, and states: **security is as strong as the weakest link**. More specifically we will first present the structure of the GSM network as well as the protocols that define it. Next we will present the SDR systems in combination with the Open Source Mobile Communication projects that we will use to carry out specific Um interface attacks. In the Hands on part of the master thesis we will integrate the limesdr mini to openBTS in order to carry out active attacks on GSM Um interface. Finally, by using HackRF, we will show how we can force our target device to downgrade from 4G to GSM technology.

Περίληψη

Καθώς παρατηρείται αυξητική τάση στις LTE και 5G τεχνολογίες, η δεύτερη γενιά κυψελωτών δικτύων GSM είναι ακόμη ενεργή και δεν έχει αποσυρθεί από πολλούς παρόχους τηλεπικοινωνιών σε όλο τον πλανήτη, γεγονός που εγκυμονεί πολλά ερωτήματα. Η πραγματικότητα είναι οτι η GSM τεχνολογία υπάρχει και θα υπάρχει για πολλά χρόνια ακόμη και θα συντηρούν οι περισσότεροι πάροχοι τηλεπικοινωνιών. Σύμφωνα με την Informa, υπολογίζονται περίπου 4.6 δισεκατομμύρια GSM subscribers συνολικά στον πλανήτη και αυτό δημιουργεί το βασικότερο ερώτημα κατά πόσο ασφαλής είναι μια ~20 χρονών τεχνολογία η οποία εξυπηρετεί τόσο μεγάλο πλήθος συσκευών. Το επόμενο ανησυχητικό ερώτημα είναι κατά πόσο εύκολα μπορεί καποιος επιτιθέμενος να προκαλέσει στον στόχο του downgrade σε GSM, από τα φαινομενικά άτρωτα LTE δίκτυα. Αυτά τα ερωτήματα θα προσπαθήσουμε να απαντήσουμε θεωρητικά, αλλά και πρακτικά, στην παρούσα πτυχιακή εργασία, καθώς και να αποδείξουμε οτι όσο υπάρχει η υπδομή των GSM δικτύων, η ασφάλεια που παρέχεται από τα νέα LTE δίκτυα μπορεί να παρακαμφθεί. Με άλλα λόγια, θα αποδείξουμε τον αλάνθαστο νόμο της ασφάλειας, που ισχυεί και για τα mobile networks, και λέει: **security is as strong as the weakest link.** Πιο αναλυτικά στην παρούσα διπλωματική εργασία αρχικά θα παρουσιάσουμε τη δομή του GSM δικτύου καθώς και τα πρωτόκολλα που το καθορίζουν. Στη συνέχεια θα παρουσιάσουμε τα SDR συστήματα σε συνδυασμό με τα Open Source Mobile Communication projects που θα χρησιμοποιήσουμε για να διεξάγουμε συγκεκριμένες επιθέσεις στο Um interface της GSM τεχνολογίας. Στο Hands-on κομμάτι της εργασίας, θα πραγματοποιήσουμε integration μεταξύ του limesdr mini και του openBTS προκειμένου να διεξάγουμε active attacks στο GSM Um interface. Τέλος, με τη χρήση του HackRF θα δείξουμε πως μπορούμε να αναγκάσουμε τη συσκευή του στόχου μας να πραγματοποιήσει downgrade από 4G σε GSM τεχνολογία.

Table of Contents

SECTION A - THEORETICAL FOUNDATION

CHAPTER 0 – INTRODUCTION TO MOBILE (CELLULAR) NETWORKS

The architecture of the mobile (cellular) networks has changed a lot in the last 30 years, in parallel with its huge growth. The Advances Mobile Phone System (AMPS), developed in 1982 in the United States, was a widely used first-generation system. Second-generation cellular systems were converted to digital voice transmission to improve capacity, increase security, and enable text messaging. The Global System for Mobile communications (GSM), which development initiated in 1991 and has become the most widely used mobile phone system in the world, is a 2G system. The demand in carrying data via cellular network [31], a new standard was developed to utilize existing GSM networks and was called General Packet Radio Service (GPRS). Another variant of GPRS which was based on a different modulation technique from GSM is Enhanced Data rates for Global Evolution (EDGE). It performs higher throughput than GPRS. Both GPRS and EDGE are 2.5 Generation Cellular systems. The third generation of systems (3G) which development initiated in 2001, provides both digital voice and broadband digital data transmission services. 3G networks provide speeds of at least 2 Mbps for static or mobile users, and 384 kbps in the case of a moving vehicle. The Universal Mobile Telecommunications System (UMTS), also known as WCDMA (Wideband Code Division Multiple Access), is the fastest growing 3G system in the world. This system can deliver speeds of up to 14 Mbps in the downlink direction and 6 Mbps in the uplink direction. [31] With the increasing demand for more bandwidth, 4G networks have been developed. The two standards of 4G networks are Log Term Evolution (LTE) and IEEE 802.16 (WiMAX). The LTE standard gained popularity in 2011 and is currently the successor to UMTS 3G technology.

The resource that was lacking in 3G systems, as was also the case in 2G and 1G systems before them, is the radio frequency spectrum. Governments licensed the right to use parts of the spectrum to mobile operators, often through a spectrum auction where they bid. Obtaining a piece of the licensed spectrum makes it easier to design and operate the systems, as no one else is allowed to transmit in that spectrum, but it often costs a lot of money. For example, in England in 2000 five 3G licenses were auctioned off. for a total of $ 40 billion.

It is this limitation of the spectrum that led to the design of the cellular network. In order to manage radio interference between users, the coverage area was divided into cells. In each cell, channels are assigned to users that do not interfere with each other. cause many interferences in neighboring cells. This allows good reuse of the spectrum, frequency reuse in neighboring cells, which increases the network capacity. In 1G systems, which carried each voice call to a separate frequency band, the frequencies were carefully selected so as not to conflict with neighboring cells. In this way a given frequency could only be reused in some cells. Today's 3G systems allow all frequencies to be used in each cell, but in a way that results in a tolerable level of interference with neighboring cells. There are variations of cellular design, which include the use of directional or sectional antennas in cell towers to reduce interference, but the basic idea remains the same.

From a statistical insight of Informa, until 2014 there were around 4.6 billion devices subscribed to GSM networks and still they continue to grow rapidly. Nowdays with the arrival and evolution of LTE and 5G technologies, GSM networks are considered as an alternative mobile communication system that is available everywhere around the world. [2]

According to International Communication Union (ITU) [30], at the end of 2020, around 85 percent of the global population is covered by a 4G network and 93% of the global population is subscribed to a mobile-broadband network. In the below chart the incremental percentage of population coverage by type of cellular network is presented for the year 2020. The values for 2G and 3G networks show the incremental percentage of population that is not covered by a more advanced technology network:

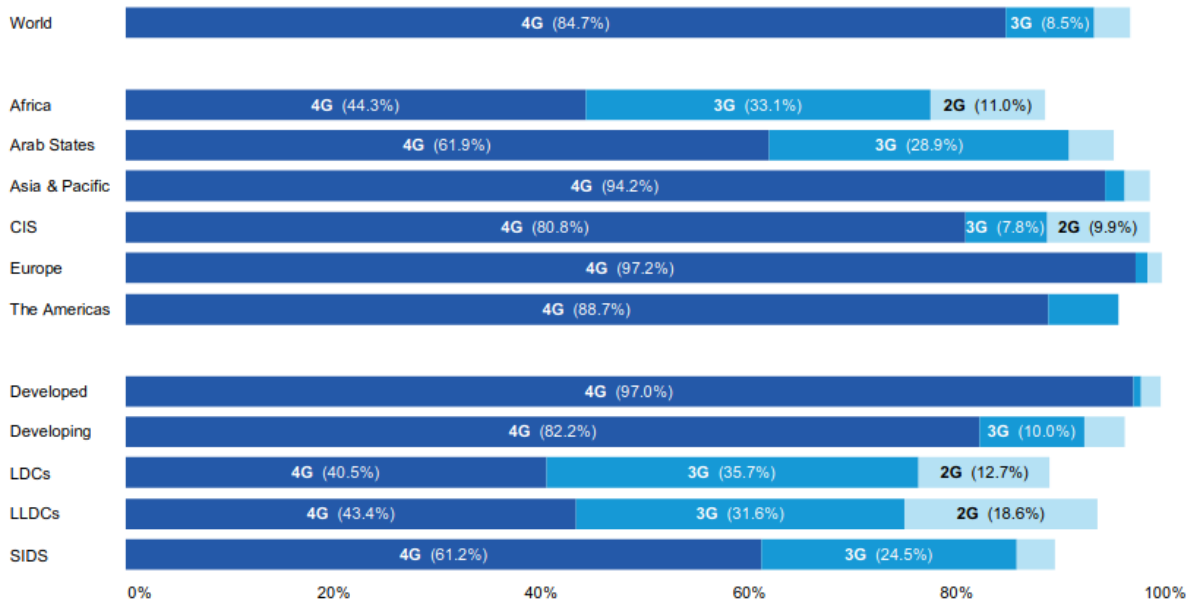

Figure 0.1 - Population coverage by type of mobile network, 2020

Why GSM?

Considering the increasing and evolution of LTE and 5G networks the question that arise is why GSM remains such a popular technology until today. This question comes to answer Kai Sahala, head of mobile marketing at Nokia. Specifically, he announced:

“New networks, such as 3G and LTE, have spotty coverage at the start and grow their coverage from urban areas. Not many countries have 100 per cent 3G or LTE coverage. So GSM can be the underlying layer for newer networks,”

In other words, GSM technology ensures a continuous robust service which provide the same functions as LTE networks but in a slower perspective. The lower frequencies of GSM networks provide much wider geographical coverage and require less physical infrastructure like cell towers.

Another reason that GSM technology is still in maintenance mode, despite that some telecommunication system providers decommissioned their GSM networks, is that GSM systems are shared with 3G/UMTS infrastructure.

As telecommunication providers deploying LTE networks, part of this work is to parallel modernising GSM networks. With those actions the main goal is to achieve hardware support for all network technologies.

According to ITU [30], for the year 2020, almost all the urban areas, globally, are covered by a next generation mobile network, it is observed that there are many gaps in rural areas. For example, in Least Developed Countries (LDCs), 17% of the population on rural areas has zero cellular coverage, and only 19% of the rural population is covered by 2G cellular networks.

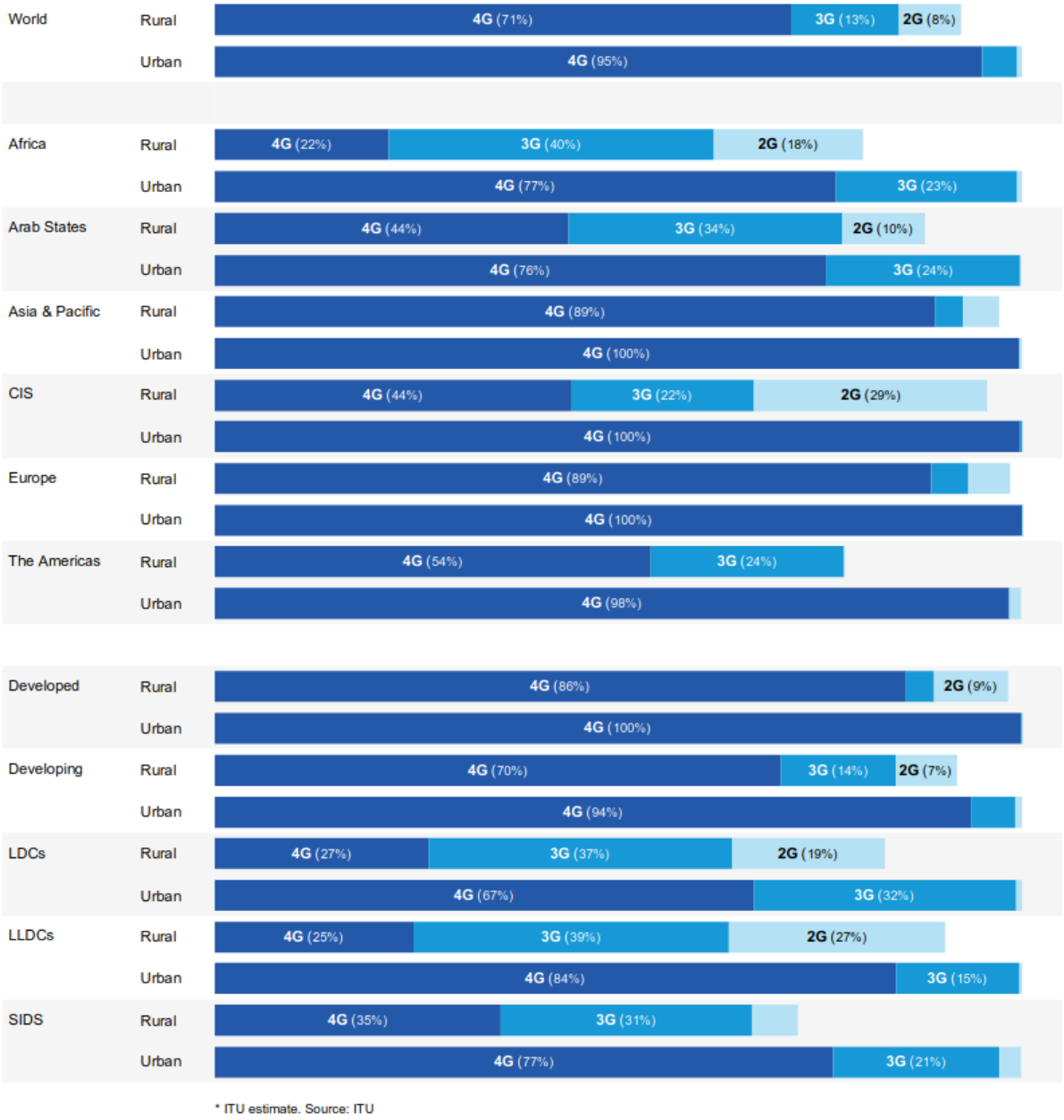

Figure 0.2 - Population coverage by type of mobile network and area, 2020

Security Aspects in GSM

As more and more subscribers use the GSM networks, the need of evolving the security elements (Authentication/Encryption) was a critical priority for the network engineers. Suddenly new attacks came up to light such as eavesdropping, impersonation, and personal data grubbing techniques due to the popularity of GSM. As a result, the corresponding countermeasures had to be applied.

Until today there is still a serious number of vulnerabilities that afflict the GSM networks. The GSM security issues can be categorized to the following sections: [1]

- **Category 1**: Theoretical vulnerabilities that practically cannot be exploited yet.
- **Category 2**: Vulnerabilities that practically can be exploited but the process is it not known or it needs sophisticate and expensive gear that are not available to the people in general.
- **Category 3**: Vulnerabilities that practically can be exploited with available equipment to the people in general.

Taking into account the scope of this master thesis, only the radio interface vulnerabilities will be mentioned and associated with Category 2 and 3:

- **Encryption in Um interface is not usually required**: As we will discuss later in this thesis, the core nodes of the GSM network activate the encryption process. Due to the fact that some old mobile devices do not support encryption, the encryption mode cannot be activated and all signalling and voice calls are transmitted plaintext. This flaw leads intelligent rogue systems to perform malicious attacks such as eavesdropping.
- **Bypassing Authentication process**: In GSM networks the mobile devices subscribe to the BTS with the higher transmission power. Therefore, attackers can deploy rogue BTS stations close to the victim's mobile station in order to perform a better transmission power for the victim's mobile station which in turn will automatically select the rogue BTS and the IMSI will be transmitted through the Location Update Request. The attacker can then use the victim's IMSI in order to perform a MitM attack between the victim and the legitimate GSM network.
- **A5/1 Passive attacks**: Sylvain Munaut and Karsten Nohl practically proved that passive attacks on A5/1 ciphering algorithm are possible under the following conditions:
  – A correctly received data stream can be sniffed from radio interface.
  – Empty bits in GSM signalling frames (fillbits) are sent with a repeating bit pattern
  – A precomputed decryption table with a size of around 4 TB to be used with the right tools.
- **A5/2 ciphering algorithm weaknesses**: It was created to allow the export of GSM systems to countries for which export restrictions concerning security technologies exist. With the processing power of today's computers, it is possible to retrieve the ciphering key Kc within seconds with only little ciphering data collected. As A5/2 is not used in countries where no export restrictions apply, this in itself is not an issue.
- **Location leaks on the GSM air interface:** Denis Foo Kune, John Koelndorfer, Nicholas Hopper and Yongdae Kim, in their paper **Location leaks on the GSM air interface**, they explore techniques in order to check if a user is present in a small area or absent from one large area, just listening to GSM broadcast channels. With a combination of readily available material and open source software, such as osmocombb, they performed practical site test attacks involving bypassing the temporary ID designed to protecting the identity of the end user. [31]
- **Denial of Service Attacks on Um Interface:** Christoforos Ntantogian, Grigoris Valtas, Nikos Kapetanakis, Faidon Lalagiannis, Georgios Karopoulos and Christos Xenakis, in their paper **Attacking GSM Networks as a Script Kiddie Using Commodity Hardware and Software**

Perform a stealthy Denial of Service (DoS) attack to a targeted, they performed a stealthy Denial of Service (DoS) attack to a targeted mobile station. The result of this attack was that the victim mobile station could not receive legitimate phone calls, without causing any suspicious activity to the victim. [32]

Passive attacks to GSM technology can be mitigated by using the A5/3 encryption algorithm for communication which at the time of writing this master thesis is considered to be secure. Today, A5/3 is supported by most of the mobile devices that are available but not for all GSM networks. Furthermore, the mobile device must not support A5/1 or A5/2 algorithms in order to prevent an attacker to calculated the ciphering key, kc.

At this point, it is worth mentioned that the above GSM passive attacks were targeted to compromise the ciphering key Kc. No practical methods are still known that with remote access to the SIM card the secret key Ki can be compromised. This means that should an attacker get the ciphering key Kc of his victim's mobile device, he would still not be able to authenticate during the next network challenge. In this way if the network requires authentication and new ciphering procedure for each communication session it will be infeasible for the attacker to conduct attacks such as caller impersonation in order to conduct or receive calls on behalf of victim.

CHAPTER 1 – GSM NETWORK ARCHITECTURE

A GSM network is divided into three subsystems which are described in the following sections: [1]

- **The Base Station Subsystem (BSS)**
- **The Network Subsystem (NSS)**
- **The Intelligent Network Subsystem (IN)**

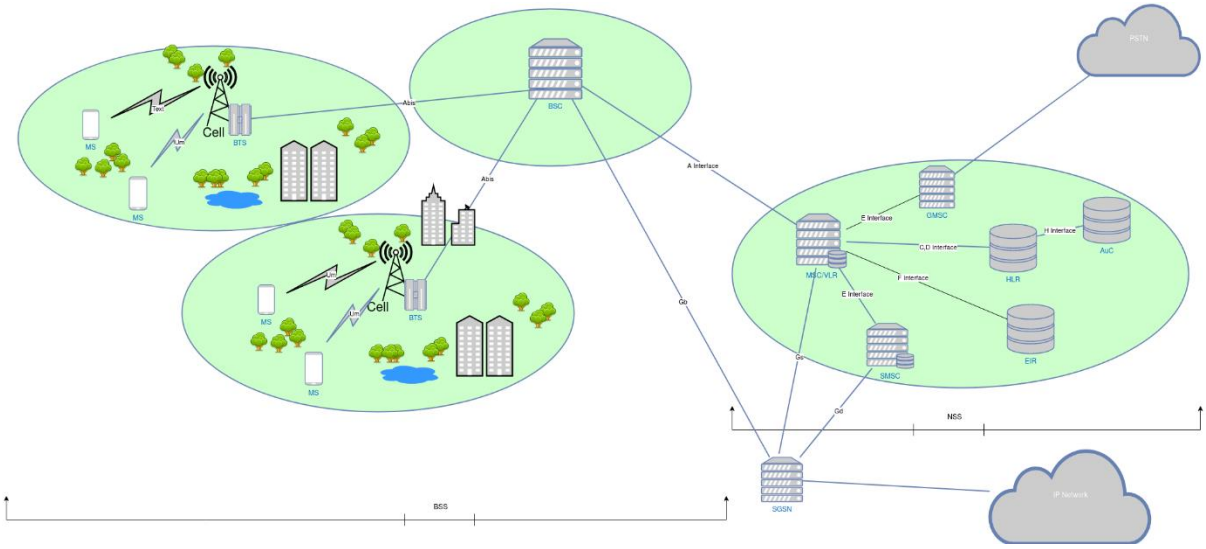

Figure 1.1 – GSM Network Architecture

1.1 The Network Subsystem

It is also called the 'core network' and is responsible to maintain functionalities such as call establishment, call control, subscriber management and call routing between mobile switching centers and other different type of networks. Other types of networks are Public Switched Telephone Network (PSTN), Voice other Internet Protocol Networks (VoIP) and international fixed-line networks.

1.1.1 The Mobile Switching Center (MSC)

The most centralized node of the core network is the Mobile Switching Center (MSC). It is responsible to manage and route all connections between subscribers. More specifically some functionalities and responsibilities of the MSC are:

- Registration of Mobile Devices when they are switched on.
- Call establishment and call routing between two subscribers.

- Routing SMS messages.
- Mobility Management of subscribers as they roam everywhere in the network.

In order the MSC to communicate with other nodes of the network, standardized interfaces are used as shown in Figure 1.1.

1.1.2 The Visitor Location Register (VLR)

Every MSC node allocates a Visitor Location Register database, which keeps information records for each subscriber that is in the restriction of a particular MSC. These records are copies of the original records which are stored in the HLR data base. The main task of the VLR is to reduce the signalling process between the MSC and the HLR. For example, if a subscriber roams into the area that is served by an MSC, the data records are copied from the HLR to the VLR associated with the particular MSC and are used for connection establishment as long as the subscriber remains in these area. At the time the subscriber leaves the coverage area of the particular MSC and roams to another MSC area, his record will be copied from the HLR to the VLR of the new MSC. His data will be deleted from the previous VLR.

1.1.3 The Home Location Register (HLR)

The subscriber database stores for every subscriber a record which contains information about subscriber's available services. The HLR also stores the International Mobile Subscriber Identity (IMSI) of each subscriber that is used as an international unique number that identifies a subscriber. The IMSI is usually a 15-digit number and is consist of:

- The Mobile Country Code (MCC): The first 3 digits represent the subscriber's home country.In Greece the MCC is 202.
- The Mobile Network Code (MNC): The next two or three digits (2-digits in Europe – 3-digits in North America) represent the subscriber's national telecommunication provider identification. Below are shown the MNC for each telecommunication provider in Greece:

Greece		
mcc	**mnc**	**Provider**
202	02	Cosmote
202	01	Cosmote
202	14	CyTA Mobile
202	04	Organismos Sidirodromon Ellados (OSE)
202	03	OTE
202	10	Tim/Wind
202	09	Tim/Wind
202	05	Vodafone

- The Mobile Subscriber Identification Number (MSIN): The remaining digits that consist the IMSI identify uniquely a subscribe within his home network and depending on MNC it is usually 9 or 10 digits long.

1.1.4 The Authentication Center

An important node that HLR is associated with is the Authentication Center (AuC) which stores the Ki for each subscriber. The same key can only be found inside the SIM card of the subscriber's Mobile Device and it cannot be retrieved. The secret key ki is used to authenticate a subscriber during a call establishment. The authentication process which is initiated during a signalling connection establishment between the subscriber and the network is described in the following steps and it is shown in figure 1.2:

- ✓ An authentication triplet is requested by the MSC from the HLR/AuC.
- ✓ Based on the subscriber's IMSI the Ki of the subscriber and the authentication A3 algorithm is retrieved by the AuC.
- ✓ The AuC uses the Ki and a rand number (128-bit) RAND as an input of the A3 authentication algorithm in order to generate the SRES (32 bits).
- ✓ The AuC also uses the Ki and the RAND as an input of the A8 algorithm in order to generate the ciphering key KC. The Kc is used to cipher the traffic connection once the authentication process has been completed.
- ✓ The MSC retrieves the authentication triplet (SRES, Kc, RAND) from HLR/AuC in order to perform the authentication of the subscriber. The Ki key never leaves the AuC.
- ✓ The MSC sends to the subscriber's mobile device an authentication request which contains the RAND.
- ✓ The mobile devices forwards the RAND to SIM card, which uses the RAND and the Ki key as an input to the A3 Algorithm in order to generate the same response SRES*. The Ki key never leaves the SIM card.
- ✓ The MSC retrieves back from the mobile device an authentication response which contains the SRES*.
- ✓ The MSC compares SRES and SRES*. If the value is the same, the authentication process is completed successfully and communication is made.

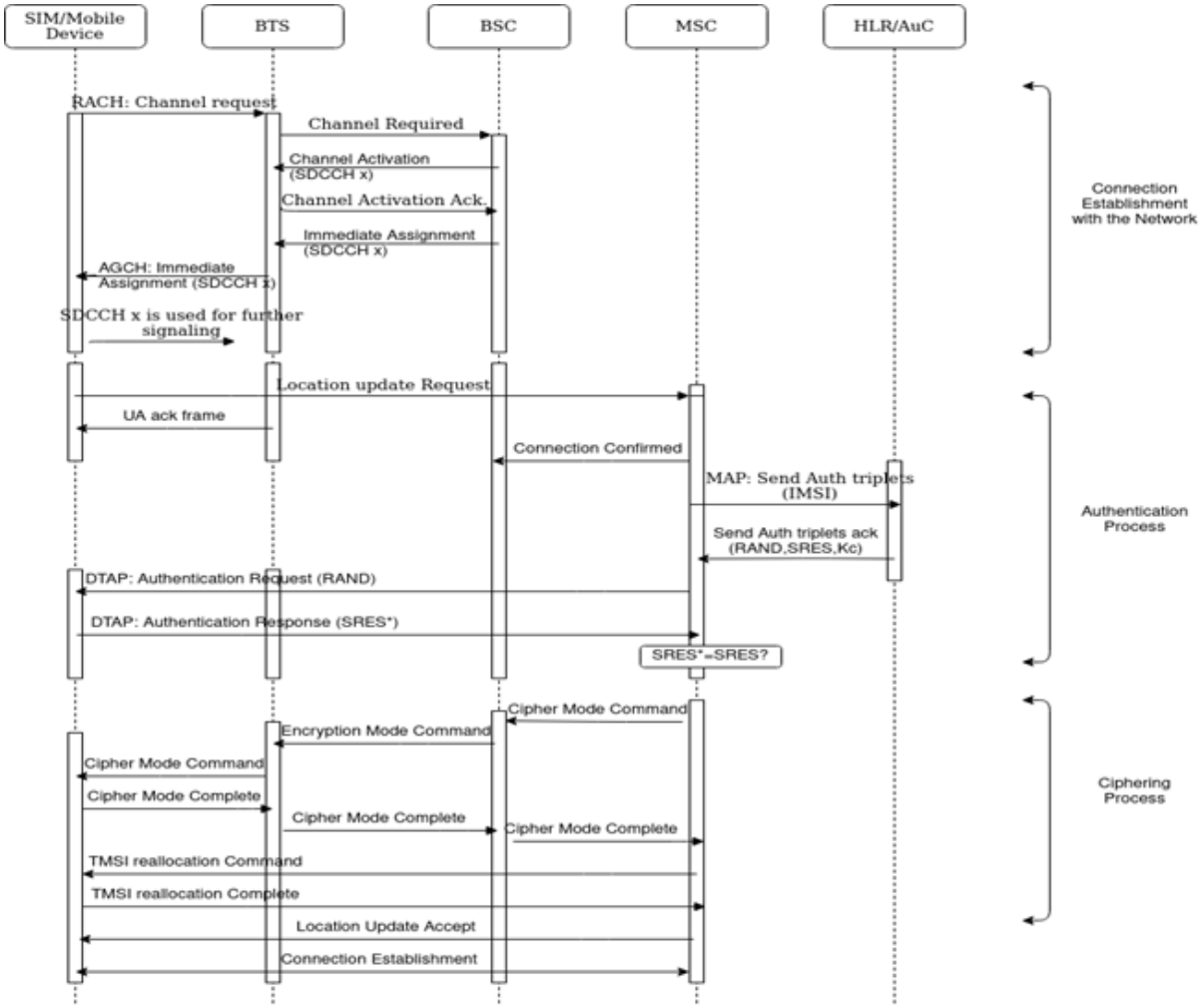

Figure 1.2 – Location Update Request Procedure

1.1.5 The Short Messaging Service Center (SMSC)

The SMSC node is used to store and forward sms messages. Some of the uses of SMS messages are chatting between subscribers and also used for the provider's services as notification alerts.

A simple routing flow of an SMS goes as follows:

- ✓ The MSC retrieves an SMS from the subscriber via a signalling channel.
- ✓ Transparently the SMS is forwarded to the SMSC.
- ✓ In order the message to be delivered to the destination, SMSC check the MSISDN of the recipient and fetches the MSC number that servers the recipient's mobile device from the HLR.
- ✓ The SMS is then forwarded to the corresponding MSC.
- ✓ When the new MSC that servers the recipient's mobile device get the SMS it alerts the MS and if it gets a respond then it forwards the SMS to the mobile device.
- ✓ When MSC receives a receipt confirmation from the mobile device, it notifies the SMSC and the SMSC deletes the SMS from its storage.

- ✓ When the contact with the mobile device is impossible due to bad signal reception of the mobile device or it is switched off, then a waiting flag message is set in the VLR and the SMS is queued in the SMSC.
- ✓ When connection with the mobile device is established, the MSC notifies the SMSC to retry delivering the SMS.

1.2 The Base Station Subsystem (BSS)

The radio network is responsible to perform wireless connection between Subscribers and the core network. The radio interface of the BSS is also called Um or air interface.

1.2.1 GSM Frequency Bands

In most regions like EMEA and APAC the GSM Band that is used is GSM-900. The downlink direction bandwidth is from 935 to 960 MHz and the uplink direction bandwidth is from 890 to 915MHz. The uplink is the communication direction from the mobile device to the radio network and the downlink is the communication direction from the radio network to the mobile device. That makes a bandwidth of 25MHZ for each direction which contains 125 channels with a bandwidth of 200KHz each as shown in figure 1.3.

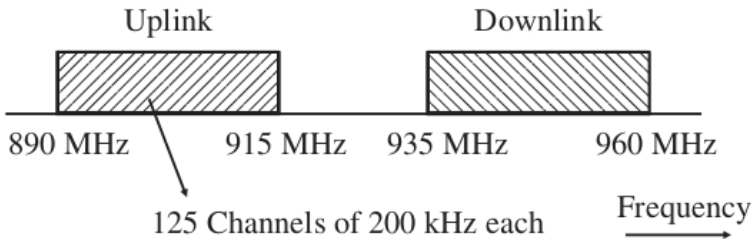

Figure1.3 – GSM900 Band [1]

As the GSM technology in Europe became more popular and the number of subscribers was rising, the need for more bandwidth was obvious. Therefore, a new frequency band was deployed by the regulating authorities and that was the GSM 1900 Band in which the downlink direction bandwidth is from 1805 to 1880 MHz and the uplink direction bandwidth is from 1710 to 1785 MHz. So for the new 1800MHz Band the bandwidth is 75MHz which corresponds to 375 additional channels as shown in Figure 1.4.

Band	ARFCN	Uplink (MHz)	Downlink (MHz)
GSM 900 (primary)	0–124	890–915	935–960
GSM 900 (extended)	975–1023, 0–124	880–915	925–960
GSM 1800	512–885	1710–1785	1805–1880
GSM 1900 (North America)	512–810	1850–1910	1930–1990
GSM 850 (North America)	128–251	824–849	869–894
GSM-R	0–124, 955–1023	876–915	921–960

Figure 1.4 – GSM Bands [1]

1.2.2 The Base Station (BTS)

The most visible node in GSM network is the Base Station which provides to each subscriber wireless connection with the core network. The area that covers a BTS is called a cell. As there are more subscribers in a particular area that need to be served simultaneously, cell can be shrink in a manner that the frequency bands to be re-used, especially in dense urban areas. Therefore, regarding the density of a particular area the radius of a cell can be from ~3 km to only about 100m. As the radius of a cell is reducing, it is also reducing and the transmitting power. So, in order the same frequency ranges to be re-used and at the same time to not interfere with each other the neighbouring cell must transmit on different frequencies and cell structures to be deployed as shown in the figure 1.5 bellow.

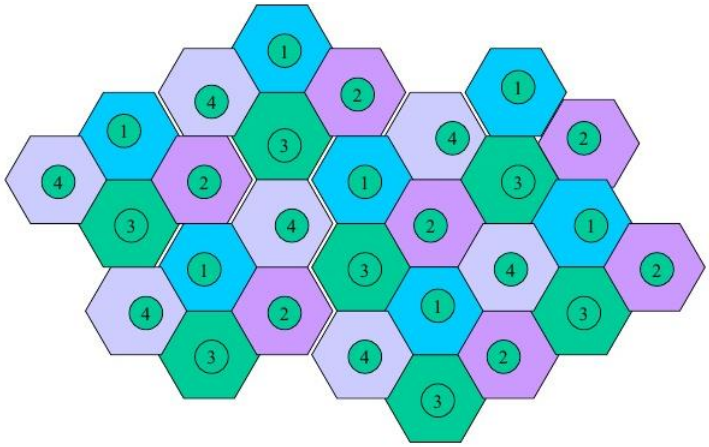

Figure 1.5 – Cell Structure K=4 [25]

Every number in a cell uses a different frequency. The frequency re-use distance of a cell structure is calculated from the cluster size K which is the offset from the center of the adjacent cluster.

Moreover, in order the capacity of a base station to be increased, the coverage is split into two or three sectors which are then run on different frequencies by the same tower. This actualization provides a better reuse of the available frequencies as shown in figure 1.6.

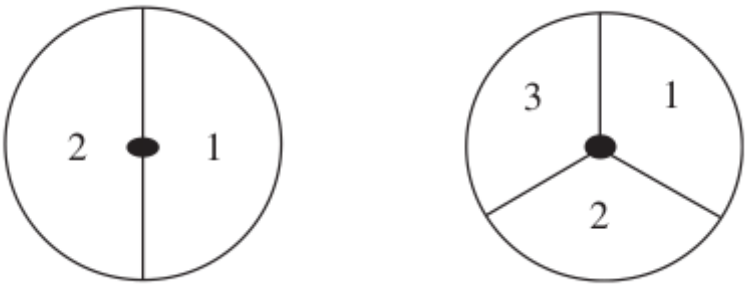

Figure 1.6 – Cell Sectors [1]

Ciphering in BTS

The ciphering process uses a stream cipher algorithm in order to encrypt or decrypt the data frames as shown in the figure 1.2. The ciphering and deciphering process is performed in BTS. As we already saw, in order a frame stream to be encrypted the below steps must be performed:

- ✓ A Kc ciphering key is generated from the A8 Algorithm which uses as inputs the RAND sequence and the key. The kc is generated in the AuC node from the network side and accordingly in the SIM card from the Mobile Station side.
- ✓ Using the GSM frame number, which is increased for every air interface frame, Kc is used as input for the A5 ciphering algorithm.
- ✓ A 114-bit sequence which is XOR combined with the bits of the plain data stream, the A5 ciphering algorithm computes the ciphering data stream.

The ciphering of a frame can be shown in the below figure.

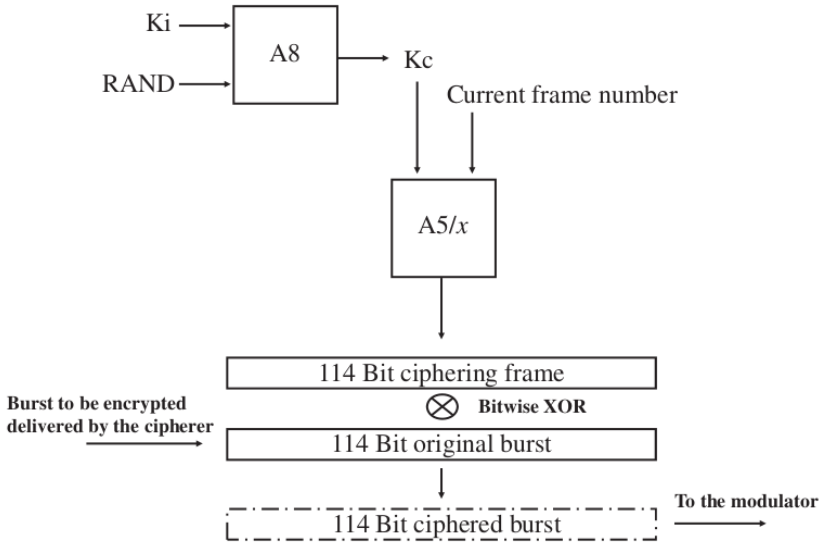

Figure 1.7 – Ciphering of a Frame[1]

There are some variants of the A5 algorithm that are available on GSM networks. The most popular different variants are A5/1, A5/2 and A5/3 algorithms. In most cases the selection of ciphering algorithm depends and on the specifications of the mobile station. So during a connection establishment the network is informed by the mobile station about the ciphering algorithms that are available to it and the network have then to choose an algorithm that is supported by the network and the mobile station.

After the Authentication process is completed, the MSC usually initialises the ciphering process by sending a ciphering command to the Mobile Station. Such message contains information about the ciphering process and the ciphering key Kc which removed and stored on BTS, before the BSC forwards the message to the Mobile Station. The BTS uses the Kc to cipher the Um interface.

1.2.3 Multiple Access

A major issue that arises when simultaneous subscribers are communicate with the BTS is the frequency capacity exhaustion. In order to overcome this issue a method is used which is called Frequency Division Multiple Access (FDMA). Furthermore, the subscribers communicate with the BTS on different frequencies. The second method that is used is Time Division Multiple Access (TDMA). In every 200KHz bandwidth channel of 125 total channels, 8 subscribers have the ability to communicate simultaneously with the BTS as shown in figure 1.8.

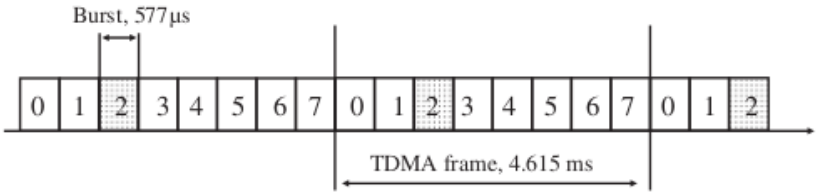

Figure 1.8 – TDMA Timeslots [1]

More specifically, the carrier frequency is time divided into frames of 4.615ms. Each frame consists of 8 timeslots, each of one is allocated for a different subscriber. The time frame of a timeslot is called a burst with a duration of 577 microseconds. For example, if the timeslot 2 is allocated to the mobile device of the subscriber x during a voice call, then the mobile device must send and receive during that burst and then have to wait for the next frame for the burst 2 in order to receive or transmit.

In TDMA method every burst is split into 6 divisions as shown in figure 1.9

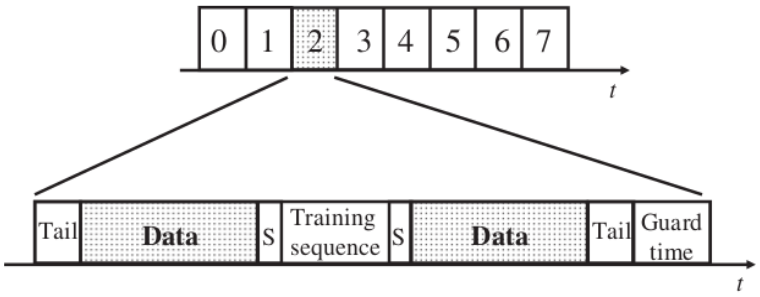

Figure 1.9 – Burst[1]

At the beginning and at the end of every burst there is a section called Tail. This helps the receiving device to recognise the beginning and the end of the burst.

In addition, each burst is separated with each other with an empty section called Guard time. This section is helpful when different subscribers are moving along the coverage area and have an ongoing changing distance from the BTS. Therefore, a signal from a subscriber who is far away from the BTS will take longer time to reach the BTS in comparison with a signal from a subscriber who is closer to the BTS. So in order to reduce eliminate the overlapping, Guard time sections are used.

Other physical factors that create issues to signal propagation are reflection and absorption of a signal. To eliminate these factors a burst contains a section called training sequence and is located in the middle of it. Training sequence contains the same bit pattern and it is used to compensate for interference caused from the above factors. At the receiving device these factors are encountered by comparing the received signal with the training sequence and thus adapting the analog filter parameters for the signal. The filter parameters, calculated this way, can then be used to modify the rest of the signal and thus to better recreate the original signal.

So what is left on the burst are two data sections with a length of 57 bits each for the subscriber's actual data which usually are a digitized voice or an SMS.

Finally, each burst contains 2 bits, one at the beginning of the training sequence section and one bit at the end of it, which are called 'stealing bits'. These bits indicate if the data sections contain data like a digitised voice call or are used ('stolen') for urgent signalling information.

1.2.4 Logical Channels

In order the subscriber's data or signalling data to be propagated, on the Um interface, Logical channels are used by grouping the physical channels which are the timeslots. On the first carrier frequency of a cell the first two timeslots are usually used for common logical signaling channels and the remaining last 6 timeslots are used for subscriber's data channels. For example, the subscriber's data timeslots are grouped into a 26-multiframe pattern. In the below figure 1.10, 8 timeslots of a frame are grouped into a two-dimension table

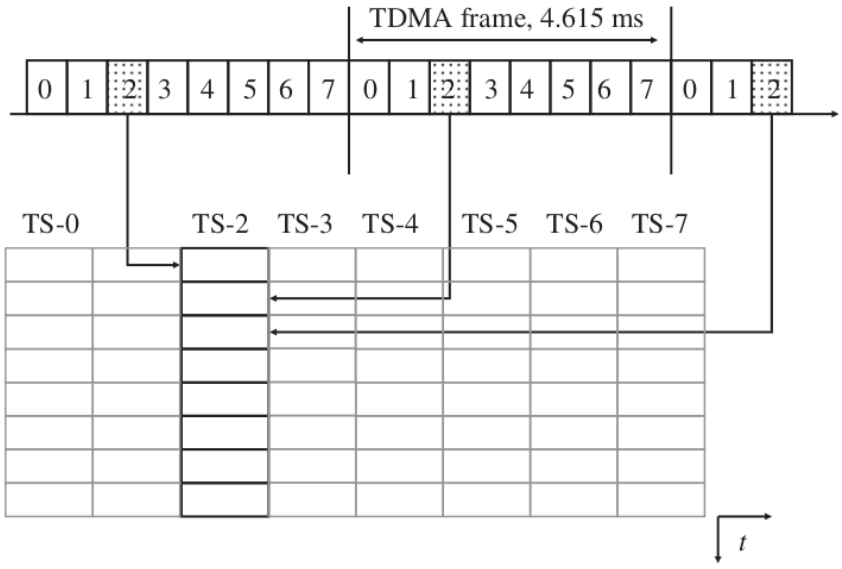

Figure 1.10 – Logical Channels [1]

Logical Channels are separated into two groups:

- **The Dedicated Channels** which are assigned to single users for their data transmission.

- **The Common Channels** which are used for the data distribution and are monitored by multiple subscribers.

The Dedicated Channels are the following:

- **The traffic channel (TCH)**: is used for subscriber's data transmission at the rate of up to 14.4Kbps. The TCH is divided into two categories which depends on their bit rate. The first category is the full rate TCH or TCH/F and the second is the half rate TCH or TCH/H.
- **The Fast Associated Control Channel (FACCH)**: coexists in the same timeslot as the TCH. The purpose of this channel is to send urgent signalling messages like a handover command. Physical bursts are not allocated for the particular channel due to the fact that its traffic is not propagated very often. To inform the mobile device that user data are removed from a TCH burst, the stealing bits to the left and right of the training sequence are used, as shown in Figure
- **The Slow Associated Control Channel (SACCH)**: is used in the uplink direction. The traffic that is transmitted from the mobile device contains information about the current cell, the neighbouring cells and signal quality measurements in order the network to be able to decide if the mobile device need to be hand over to another cell or needs the transmission power to be changed in the device. If transmission power needs to be changed, the network used the downlink direction to transmit power control commands through SACCH.
- **The Standalone Dedicated Control Channel (SDCCH)**: is a signalling logical channel. It is used for the location update procedure, for SMS transmission or during a call establishment when a subscriber has not yet been assigned a TCH.

The common channels are the following:

- **The Synchronization Channel (SCH)**: is used when a mobile device searches for the network.
- **The Frequency Correction Channel (FCCH)**: is used for transceiver calibration by mobile devices.
- **The Broadcast Common Control Channel (BCCH):** is the main channel that the mobile devices monitors and it transmits cell and network information. Some of the information that is transmitted through BCCH is the MCC, the MNC, the LAC, the cellid, and the frequencies used by the neighboring cells.
- **The Paging Channel (PCH)**: is used to notify idle mobile devices for incoming calls or SMS messages. Inside the PCH messages the most important information that is transmitted in all the cells from the LAC that the subscriber is roaming in, is the IMSI or the TMSI of the subscriber.
- **The Random Access Channel (RACH)**: is used by the mobile devices for the initial communication with the network when the mobile device receives a message via the PCH that the network is requesting a connection establishment or if the user wants to establish a call or send an SMS.
- **The Access Grant Channel (AGCH)**: If a subscriber sends a channel request message on the RACH, the network allocates an SDCCH or, in exceptional cases, a TCH, and notifies the subscriber on the AGCH via an immediate assignment message. The message contains information about which SDCCH or TCH the subscriber is allowed to use.

1.2.5 The Base Station Controller (BSC)

The most critical tasks of a BSC is the establishment control, release and maintenance of all connections for the cells that are attached to it. Furthermore, the establishment of a signaling link between the mobile device and the network and also the process of assigning SDCCH and TCH of a base station, is shown in the figure 1.2.

In other words, when a subscriber tries to perform a call or send an SMS message:

- ✓ the device sends a channel request to BSC.
- ✓ The BSC checks if an SDCCH is available and activates the channel in the BTS.
- ✓ In the next step the BSC uses the AGCH channel to send an immediate assignment message to mobile station which contains the number of the assigned SDCCH channel.
- ✓ The mobile device then uses the SDCCH to send DTAP messages that the BSC forwards to the MSC.

Similar steps of signaling channels establishment are used when the incoming calls and SMS messages try to reach the mobile station:

- ✓ The MSC node send a paging message to the BSC which contains the IMSI, the TMSI of the callee and furthermore the LAC in which the subscriber is currently located.
- ✓ The BSC holds information that uses to identify all cells in which the subscriber needs to be paged.
- ✓ At the time the mobile station receives the paging message, it replies back to the BSC with the same steps that we saw above by sending a channel request message.

The BSC is also responsible for the handover process. The handover process is when a subscriber roams through the network while a call is ongoing and happens to leave the coverage area of the cell in which the call was initially established. Then the BTS has to redirect the call to the corresponding cell

1.3 The Intelligent Network Subsystem (IN)

Additional services are provided by telecommunication service providers and for that reason additional databases and intelligent systems are needed. For example pre-paid services were introduced in the middle of 1990s were a prepaid account is associated with a subscriber and is topped up with an amount that is determined by the subscriber. Prepaid systems are also connected to the SMSC and the GPRS network. Therefore, prepaid subscribers can also be charged in real time for the use of these services.

CHAPTER 2 - MOBILE STATION – BSS PROTOCOL STACK (Um Layer)

As we already explained in Chapter 1, the Mobile station communicates with the Base Station through the air interface. There are three layers on air interface (Um) and are described below:

1. The Physical layer (Layer 1)
2. The Data Link Layer (Layer 2)
3. The Network (Message) Layer (Layer3)

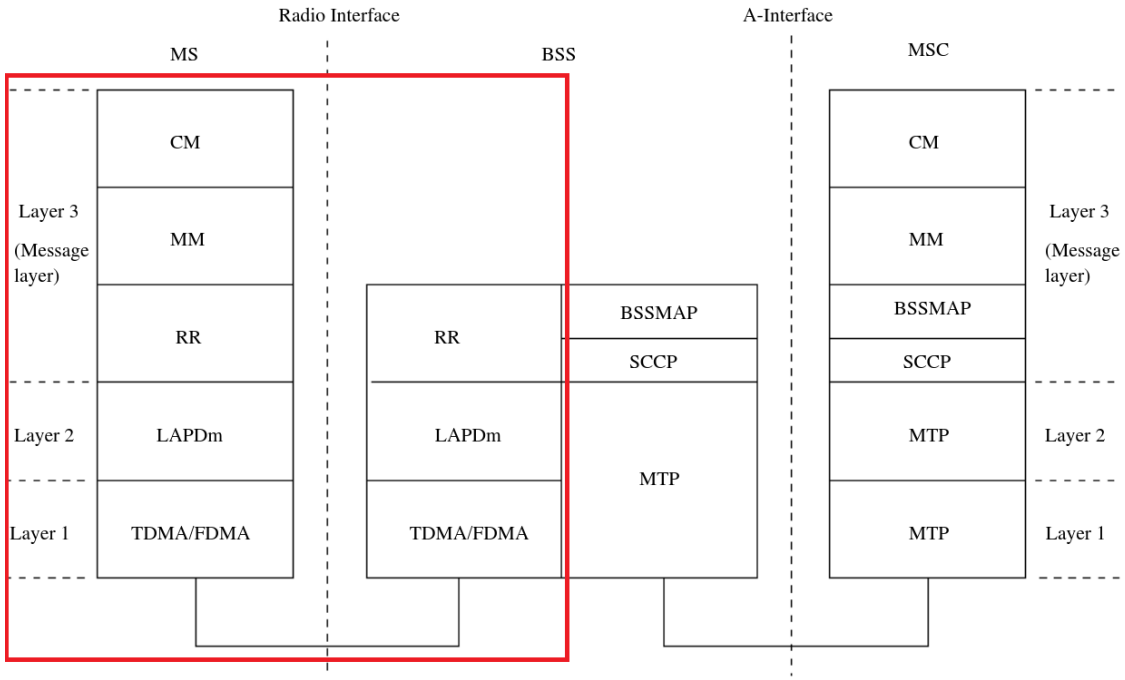

Figure 2.1 – The GSM Protocol Stack [26]

2.1 The Physical Layer

The Layer 1 is the lowest layer in the Um Interface and is consists of the all the rightful functionalities required in order bit streams to be transmitted on the physical medium. The physical medium is the air (Um) interface. As we have already shown from the network perspective the bit streams are transmitted through logical channels that are categorized in traffic channels and signaling channels. The physical layer includes the physical and logical channels that have already been covered in chapter 1.

The main purpose of the physical layer is to link the upper layers which are the Data Link Layer, the Radio Link Control and Medium Access Control layer and the supported functional units of the application as shown in the figure 2.2.

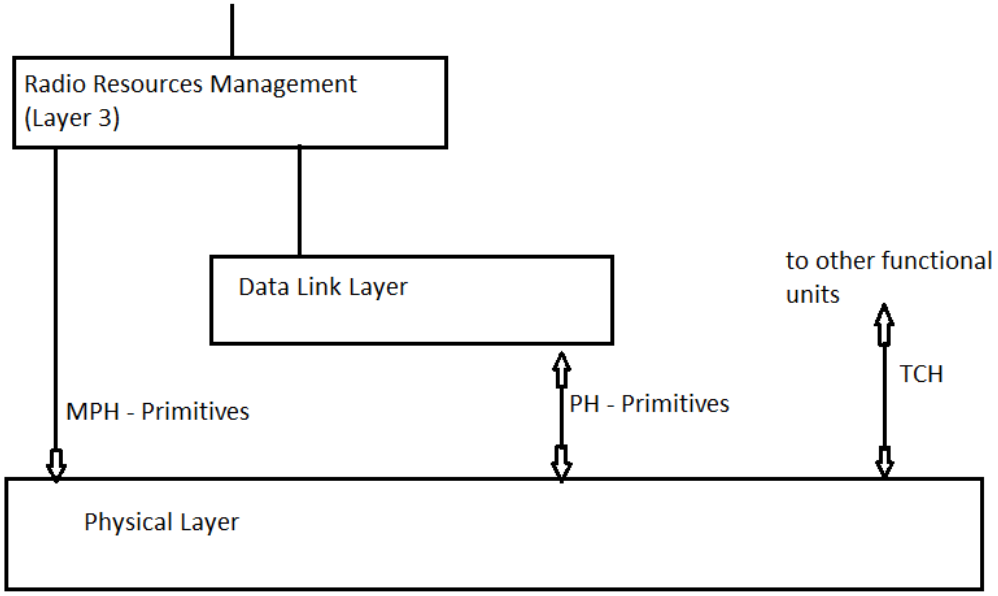

Figure 2.2 – Interfaces with the Physical Layer [27]

- The physical layer links the Radio Resource management (RR) which is a sublayer of layer 3 in the Mobile Station and in the network. The main purpose of the RR sublayer is to control the establishment and the release of the logical channels based on power measurements which allows the mobile station to select the best cell during a signaling channel establishment.
- As physical layer links the logical channels to layer 2, then it is allowed to transmit an encrypted bit streams over the air interface.
- The other functional units are used to control the channels that are specified for the data traffic.

The physical layer implements three types of procedures in order to provide functionalities to the upper layers.

- ✓ **Interface to Radio Resource Management**. These procedures are composed of MPH - primitives between the physical layer and the RR sublayer of the layer 3. The communication between the physical layer and the RR sublayer through MPH – primitives are responsible for the assignment of channels and for the transfer of various information such as measurement results.
- ✓ **Interface to the Data Link Layer**. PH – primitives are used for the communication between the Physical Layer and the Data Link Layer. More specifically they are used for the transfer of layer 2 frames and to indicate the establishment of channels to layer 2.
- ✓ **Interface to other functional units**. These procedures are responsible to interface the physical layer with other functional units in the Mobile Station and in the Network for the support of traffic channels.

2.2 Service Primitives

Service primitives is a term to specify interactions between Layers. In a high level way, they represent the logical exchange of information and control between the neighbour layers. They do not specify or constrain implementation.

We can find four types of primitives that are exchanged between the (N + 1)- layer and the (N)-layer as we can see in figure 2.2.3.

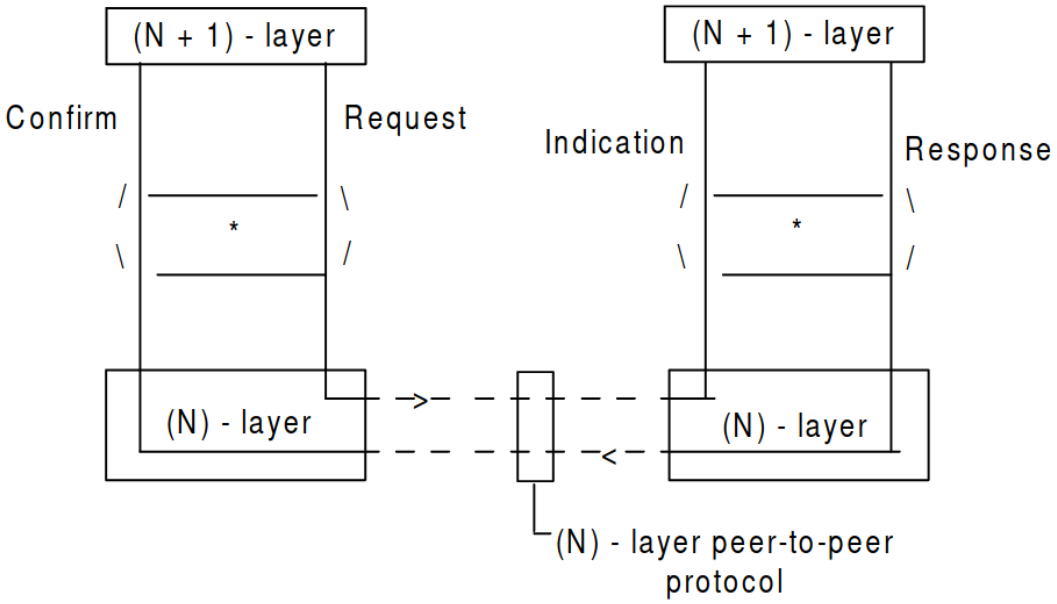

Figure 2.2.3 – Primitive Types for peer-to-peer communication [28]

- ✓ **The REQUEST primitive type**. When a higher layer requests a service from its neighbour lower layer.
- ✓ **The INDICATION primitive type**. When activities related to the REQUEST primitive type need to be taken, the INDICATION primitive type is used by a layer to notify the neighbour higher layer.
- ✓ **The RESPONSE primitive type**. When a layer needs to send an acknowledge message, from a lower layer, of the INDICATION primitive type, the RESPONSE primitive type is used.
- ✓ **The CONFIRM primitive type**. It is used for confirmation of activities completion by the layer who requested the service.

2.3 The Data Link Layer

The next layer of the GSM BSS-MS protocol stack is the Data Link Layer. It uses the signaling channels from the Physical layer to provide data link connection to the upper layer which is Layer 3. The particular procedure is performed by implementing the Link Access Procedures on the Dm channel (LAPDm) protocol. Dm is a reference name to the signaling channels. LAPDm comes from an older link layer protocol which is called HDLC and is similar to the ISDM Layer 2. The services that are provided to Layer 3 are a combination of services and functions from the physical and data link layers.

The data link layer provides services to Layer 3 through the four service primitive types that are exchanged between the data link layer and we have already discussed them in the previous section.

2.3.1 LAPDm Functions and Procedures

The main task of LAPDm protocol is to transfer information between Layer 3 objects through the Um interface by using the Dm channel. More specifically, the LAPDm protocol supports:

- ✓ Multiple layer 3 objects,
- ✓ Multiple physical layer objects,
- ✓ BCCH signalling,
- ✓ PCH signalling,
- ✓ AGCH signalling,
- ✓ DCCH signalling.

The functions of the LAPDm protocol are the following:

- ✓ Provision of one or more data link connections on a Dm channel,
- ✓ Allowing frame types recognition,
- ✓ Allowing a transparent connectivity for the layer 3 messages between layer 3 objects,
- ✓ Maintaining a sequential order of frames across data link connections,
- ✓ Providing detection of errors on the data link,
- ✓ Providing notifications for unrecoverable errors to the layer 3 object,
- ✓ Providing flow control on the data link,
- ✓ After an access request has been performed on the RACH channel, when establishing a data link it provides contention resolution.

2.4 The Network Layer (Layer 3)

The final layer of the GSM MS-BSS protocol stack is the Network Layer or the Layer 3. It consists of three sublayers which are:

- The Radio Resource Management (RR) sublayer. It provides services to the MM sublayer and utilizes the services of signalling layer 2,
- The Mobility Management (MM) sublayer. It provides common services to the entities of the Connection Management (CM) sublayer,
- The Connection Management (CM) sublayer. The CM sublayer includes, among others, the CC, SS, and SMS entities, which are independententities.

2.4.1 The Radio Resource Management (RR) sublayer

It holds the functions related to the management of the assignment, maintenance and release of logical channel connections on the radio link. It is normally terminated in the base station controller (BSC). The RR sublayer can be in two different modes. When a connection has been established, the RR sublayer is in dedicated mode. Otherwise, when a connection has not been established, it is in idle mode.

More specifically, from the MS side, when a connection is in the idle mode, the RR procedures perform reception checks and measurement of the BCCH and CCCH channels. It is performed by the MS in case the need of changing cell arises. The MS constantly measures the reception level of all the neighbouring cells. Then it checks the system information messages from each cell that it tried to synchronize and it calculate the parameters for the selection of a cell.

In order to switch from idle mode to dedicated mode, the RR sublayer on the MS must initiate an immediate assignment procedure. This procedure could be performed in two cases. The first case is when a request from the MM sublayer has been receipt. The second case is when a response to a Paging Request message assigned to IMSI and received when listening to the CCCH channel.

2.4.2 The Mobility Management (MM) sublayer

Mobility management is one of the major functions of a GSM network that allows mobile phones to work. The aim of mobility management is to authenticates users and to track where the subscribers are, allowing calls, SMS and other mobile phone services to be delivered to them. Connection management services and registration services are also provided to the various entities of the CM sublayer, or the upper layers directly. It relies on the lower RR sublayer to establish a connection between the MS and the network.

The MM sublayer consists of three type of procedures:

- **Common procedures**. It prevents an attacker to identify and locate a subscriber by performing TMSI allocation. In most cases, it is performed when the subscriber roams to another Location

Area. The procedure is initiated by the network by sending a TMSI Reallocation message to the MS. As a respond, when the MS copy the information to its SIM card, it sends to the network a TMSI Reallocation Complete message. The Authentication procedure also enables the network to request the MS to authenticate in order to check if its identity has been compromised or not.

- **Specific procedures**. They are related to the location update request procedure. The MS sends to the network a Location Updating Request message specifying the location. The network then initiates various common procedures, like TMSI reallocation.
- **Connection Management procedures**. Management services connectivity are provided by the MM sublayer to the entities of the upper CM sublayer when is requested from a CM entity. The connection management procedures are used for establishing, re-establishing, maintaining, and releasing an MM connection.

2.4.3 The Connection Management (CM) sublayer

The Connection Management sublayer contains two type of Call Control (CC) procedures which are the call establishment procedures and the call clearing procedures.

During the establishment process, the CC object on the MS initiates a CC connection establishment. This is done with the request of an MM connection from the MM sublayer. A Setup message, which contains information for the network in order to be able to process the call, is sent to the network by the CC object. In the recipient MS a notification for a new call is delivered from the network. Furthermore, a CC connection is established in order to receive the Setup message. When it is received the recipient MS answers back with a Call confirm message. If the MS is available, a Connect message is sent to the network. Then, the network will respond with a Connect Acknowledge message and it will establish the traffic channel for the two MS. The network sends a Connect message to the MS which initiate the call process.

The clearing procedure initiates when one of the two Mobile Stations sends a Disconnect message to the network. When then network receives the Disconnect message, it instantly sends a Release message to the MS that do not initiate the clearing procedure and starts the procedures to terminate the connections. A Release Complete message is sent from the second MS.

SECTION B – DEPLOYING OUR LAB ENVIRONMENT

In this section we will introduce the SDR systems and the open-source mobile communications software that will be the core of our lab environment. More specifically the section is divided into three chapters. In the first chapter we will introduce the limeSDR mini and HackRF one as the SDR systems that will be used in our LAB environment. In the second chapter we will introduce OpenBTS osmoTRX radiomodem as the open source mobile communications software. Finally, in the third chapter we will integrate the LimeSDR mini with the OpenBTS, as support for the LimeSDR mini is currently not provided by the OpenBTS.

CHAPTER 1 - SDR SYSTEMS [33] [34]

A Software Defined Radio (SDR) system is a radio system which consists of a hardware peripheral and uses software components for performing signal-processing functionalities like modulation/demodulation and others. The software component can be installed in every general purpose computer as well as most of other type of computers.

SDR systems have a major utility in mobile software network services. The hardware part of an SDR system is assembled in the front with an amplifier filter component. It is also consists of analog to digital and digital to analog converters.

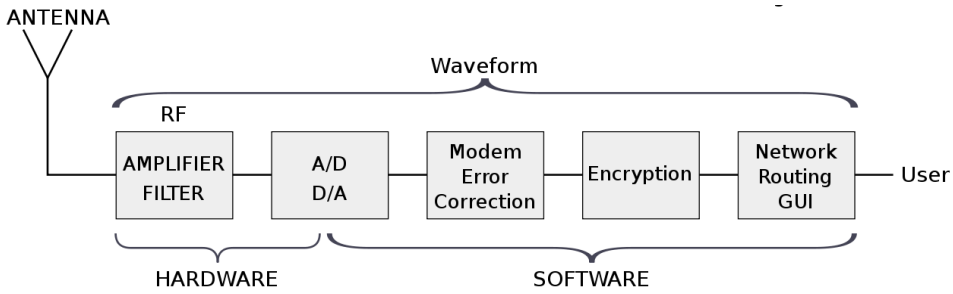

Figure 1.0 - SDR System [32]

Open source SDR systems started to be used for simple radio modem project implementations, but in near future it is expected to be used for every radio communication system, due to the following capabilities that haven't been possible before:

- SDR systems can be reconfigured in real time to any application. For example, it could be a GSM BTS tower one minute, and a GPS receiver the next.
- SDR systems can be upgraded very easy and quick.
- SDR systems can send and receive to multiple channels at the same time.

- Researchers and network engineers, with the use of SDR systems, are able to experiment, develop and test new radio technologies that have never existed before, or optimize the existent radio systems.

GNU Radio [33] [35]

As we already have discussed, an SDR system is a radio communication system which its components (mixers, filters, amplifiers, modulators/demodulators, and so on) are implemented in software on a general purpose computer, in contrast with the typical radio communication systems that their components are implemented in hardware. Software Radio brings the revolution to radio system due to its ability to create new radio system technologies and the flexible reconfiguration.

GNU Radio is a free, open source –source software development toolkit that provides signal processing blocks to implement software radios [35]. It can be used with SDR hardware platforms or in a simulation-lab environment without the need of using hardware. The GNU Radio toolkit's software components are constructed in a two-tier structure. The performance-critical signal processing blocks are implemented in C++, while the higher-level organizing, connecting, and gluing are done using Python.

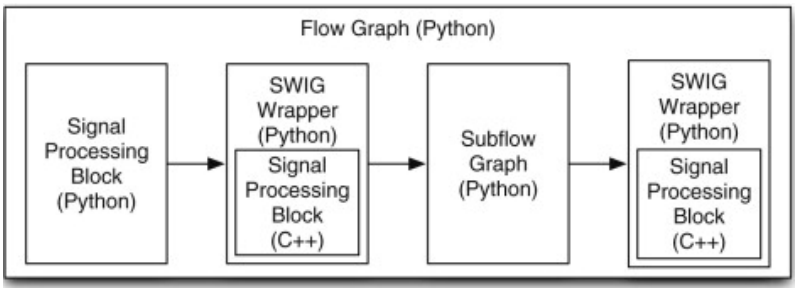

Figure 1.0.1 – GNU Radio two – tier structure [35]

1.1 LimeSDR Mini [4]

The LimeSDR Mini board is an SDR hardware component that delivers a frequency range of 10 MHz up to 3.5GHz and bandwidth of up to 30.72 MHz. The heart of the LimeSDR mini is the LMS7002M transceiver. It provides USB 3.0 port and two Rx/Tx channels. One strong feature of the LimeSDR mini is its price point, being much lower than other SDR boards with the same capabilities. At the moment of the writing of this master thesis, the cost of LimeSDR mini is about 139 US dollars.

Figure 1.1 – LimeSDR mini

1.2 LimeSuite [6] [8]

LimeSuite is a group of driver components that support SDR boards with for the LMS7002M transceiver as it LimeSDR mini. The LimeSuite consists of the following components:

- ✓ The LimeSuite library that provides C-style API
- ✓ The LimeSuiteGUI application which provides many functionalities such as accessing and modifying the low-level chip, board settings and updating the firmware
- ✓ The LimeUtil command line tool is used for listing the online LimeSDR devices connected to a computer as well as updating firmware.
- ✓ The LimeQuickTest application which performs some diagnostic checks on the hardware.
- ✓ LimeSuite API

1.3 HackRF One [5]

HackRF one is a Great Scott Gadgets SDR board that delivers a significant frequency ranges of 6 MHz up to 6 GHz and a respectful bandwidth of 20MHz. That means that the maximum sample rate that the board can perform is 20 MSamples per second. The only drawback of the HackRF board is the half-duplex transceiver. The HackRF One features a high speed USB 2.0 port.

Figure 1.2 –HackRF one

CHAPTER 2 – OPEN SOURCE MOBILE COMMUNICATION SOFTWARE

The integration of Open Source Mobile Network Projects and Software Defined Radio systems introduces a new way of how the mobile networks are constructed. All components of a classic GSM network can now be deployed as software components. The software platform that we will use as our GSM Network Lab is OpenBTS which we will cover below.

2.1 OpenBTS [9]

OpenBTS is a Linux software application that in combination with software-defined radio systems provides a standard 3GPP air interface to user devices, while simultaneously presenting those devices to the Internet as SIP endpoints.

In order to provide the aforementioned functionalities, the installation of OpenBTS requires the following software components:

- ✓ The OpenBTS component which is the actual component and performs the GSM protocol stack above radiomodem.
- ✓ The Transceiver component which is the radiomodem software implementation, controlling the SDR board.
- ✓ The Asterisk 11 component which presents the Mobile devices as SIP endpoints. It is responsible for the MSC functionalities.
- ✓ SIPAuthServer component which is the database for the subscriber information. It represents the HLR in the actual GSM network.
- ✓ SMSQue component which is the SMS forwarding server.

2.2 OsmoTRX-lms [11]

OsmoTRX is a software-defined radio transceiver that implements the Layer 1 physical layer of a BTS that provides the following 3GPP specifications:

- ✓ TS 05.01: Physical layer on the radio path
- ✓ TS 05.02: Multiplexing and Multiple Access on the Radio Path
- ✓ TS 05.04 Modulation/Demodulation
- ✓ TS 05.10 Radio subsystem synchronization

OsmoTRX is based on the Transceiver radiomodem from the OpenBTS project, but it has been configured to be compatible with a variety of open source mobile communication projects, while still maintaining backwards compatibility with OpenBTS. Currently there are numerous features contained in OsmoTRX that extend the functionality of the OpenBTS transceiver.

As at the moment of writing, the OpenBTS radiomodem Transceiver is not compatible with LimeSDR mini, we will use the osmotrx variation for LimeSDR mini as a radio modem instead, called osmo-trx-lms. We will also use the LimeSuite drivers which are applied as dependencies by the osmo-trx-lms. With osmo-trx-lms we automatically avoid using the complex driver stack with UHD, SoapyUHD and SoapySDR.

CHAPTER 3 – INTEGRATING LIMESDR MINI WITH OPENBTS

In this chapter, we will try to integrate LimeSDR mini with OpenBTS on an Ubuntu 18.04 LTS machine in order to deploy a rogue femtocell. By manipulating malicious the femtocell we will perform some active attacks later in this thesis as shown in the figure 3.1.

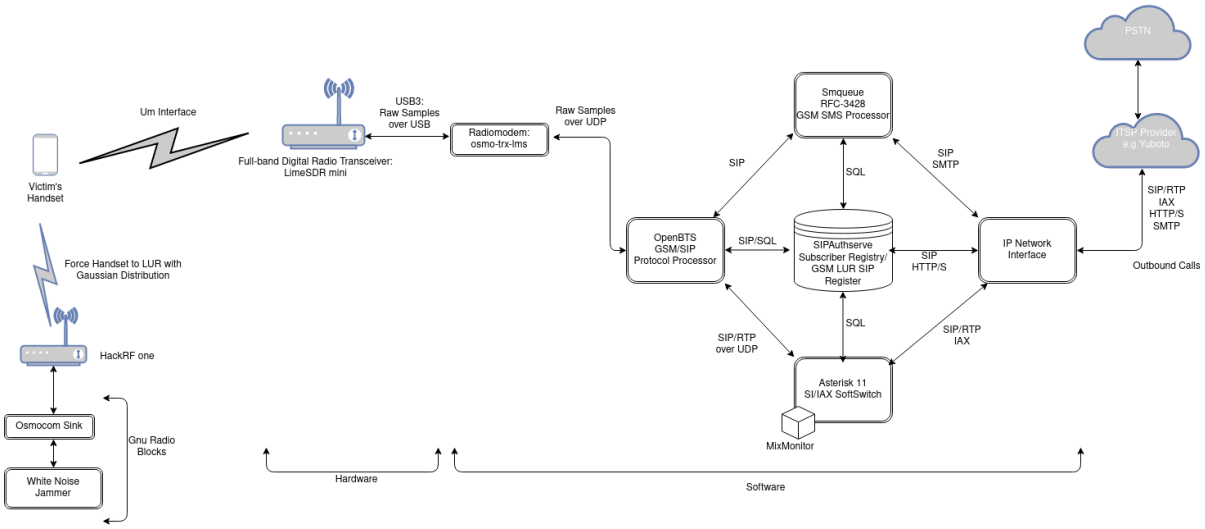

Figure 3.1 – Lab Environment

The components that we will use are:

- ✓ LimeSDR mini board
- ✓ Software radiomodem osmo-trx-lms
- ✓ LimeSuite drivers
- ✓ OpenBTS 5.0 stack full installation
- ✓ Ubuntu 18.04 LTS

3.1 Preparing the OS and installing OpenBTS [9]

OpenBTS stack can be built only on legacy versions of Ubuntu, such as 12.04, 14.04 or 16.04, using the latest OpenBTS stack 5.0. So the base OS that we will use is Ubuntu 16.04 LTS on a Vmware-based virtual machine. After the installation of the OS is finished we proceed with the necessary updates. Then, we will install git and other dependencies. To do so, on a terminal we run the following commands:

$ sudo apt update && upgrade –y

$ sudo apt-get install autoconf libtool libosip2-dev libortp-dev libusb-1.0-0-dev g++ sqlite3 libsqlite3-dev erlang libreadline6-dev libncurses5-dev

$ cd /home/rogue/dev/libcoredumper/

$ wget https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/google-coredumper/coredumper-1.2.1.tar.gz

$ sudo apt-get install software-properties-common python-software-properties

$ sudo add-apt-repository ppa:git-core/ppa

$ sudo apt-get update

$ sudo apt-get install git

The next step is to download the OpenBTS stack on our Ubuntu system. We run the following command in order to download the scripts:

$ git clone https://github.com/RangeNetworks/dev.git

Now that we have downloaded the scripts we run the following commands in order to download all the components of OpenBTS stack:

$ cd dev

$ ./clone.sh

Now every component of the OpenBTS stack will be cloned from Github in our dev/ directory. Then, in order to build the latest branch of OpenBTS, we execute the following command:

$ ./switchto.sh 5.0

Now, everything is set up for building the OpenBTS stack code. In order to compile the code, we will execute the build.sh script. It automatically installs the compiler and autoconfiguration tools as well as any required dependencies. It also controls which radio transceiver application will be built. As we have mentioned previously, OpenBTS does not support LimeSDR mini by default as the Transceiver software radiomodem is not compatible, we will first build the code as it was meant for a USPR B210 board. In order to do that we run the following command:

$ ./build.sh B210

The process can take a while and after it finishes we will have a new directory named “BUILDS” with all the necessary .deb files inside:

$ ls dev/BUILDS/2014-07-29--20-44-51/*.deb

liba53_0.1_i386.deb

range-asterisk-config_5.0_all.deb

libcoredumper1_1.2.1-1_i386.deb

range-configs_5.0_all.deb

libcoredumper-dev_1.2.1-1_i386.deb sipauthserve_5.0_i386.deb

openbts_5.0_i386.deb

smqueue_5.0_i386.deb

range-asterisk_11.7.0.4_i386.deb

Now that we have the above builds for all of our components ready to be installed, on a terminal we execute the following commands:

$ cd dev/BUILDS/2020-11-10--18-42-12/

$ sudo dpkg -i range-configs_5.0_all.deb

$ sudo dpkg -i range-asterisk*.deb

$ sudo apt-get install –f

$ sudo dpkg -i sipauthserve_5.0_i386.deb

$ sudo apt-get install -f

$ sudo dpkg -i smqueue_5.0_i386.deb

$ sudo apt-get install –f

$ sudo dpkg -i openbts_5.0_i386.deb

$ sudo apt-get install -f

We now have a fresh install of OpenBTS 5.0 stack in our Ubuntu 16.04LTS system. The next step to continue the integration process with LimeSDR mini is to delete the transceiver software from the dev/ directory [14]. To accomplish that we run the following command:

$ rm –f dev/transceiver

3.2 Upgrading to Ubuntu 18.04 LTS – Installing osmo-trx-lms

Due to the fact that we faced dependency issues in the building process of the latest branch version of LimeSuite and osmo-trx-lms on Ubuntu 16.04 LTS, we will proceed with the upgrade of the OS to version 18.04. We have tried building and installing the OpenBTS stack on Ubuntu 18.04 LTS directly but the process failed because of missing dependencies. So, the gap between OpenBTS stack and the latest version of Limesuite and osmo-trx-lms is obviously a dependency issue and that is the reason why we first built and install the OpenBTS stack on an Ubuntu 16.04 and then upgrade it to 18.04.

So, we will upgrade the OS to 18.04 version by executing the following command:

$ sudo apt update & sudo apt upgrade & sudo apt dist-upgrade –y

After the upgrade is finished, the next step is to proceed with the installation of the latest LimeSuite by cloning from github or by simply executing in a terminal the following commands[22]:

$ sudo add-apt-repository -y ppa:myriadrf/drivers

$ sudo apt-get update

$ sudo apt-get install limesuite limesuite-udev

The next step is to connect the LimeSDR mini to a USB3 port and check if it recognized properly by the drivers. Keep in mind that we need to provide USB 3.0 port support during the vmware virtual machine set-up, otherwise the limesdr mini won't work:

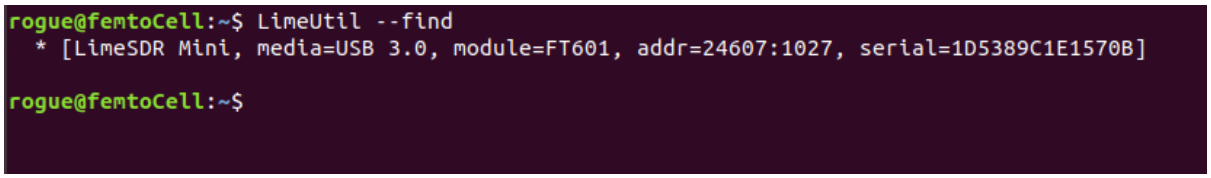

Figure 3.2.1 – Identifying LimeSDR Board

We can upgrade the firmware by executing the below command:

$ LimeUtil --update

Now, that the drivers have been configured properly, we will proceed with the installation of the radiomodem. Keep in mind that we have already deleted the radiomodem transceiver and we are going to use osmo-trx-lms instead. We can install the latest version of osmocom-trx by cloning it from github or by simply running the following commands:

$ wget
https://download.opensuse.org/repositories/network:/osmocom:/latest/Debian_10/Release.key

$ sudo apt-key add Release.key && rm Release.key

$ echo "deb
https://download.opensuse.org/repositories/network:/osmocom:/latest/xUbuntu_18.04/ ./" >
/etc/apt/sources.list.d/osmocom-latest.list

$ sudo apt-get update

$ sudo apt-get install osmo-trx-lms

So now that we've got all the pieces of the puzzle, it's time to connect the SDR to Osmo-TRX-LMS and connect Osmo-TRX-LMS to OpenBTS. One last step remains in order to finish the compatibility process. We need to adjust two settings in the osmo-trx-lms.cfg config file of the osmo-trx-lms radiomodem:

$ cd /etc/osmocom

$ sudo vi osmo-trx-lms.cfg

We adjust the two settings as shown in the below figure:

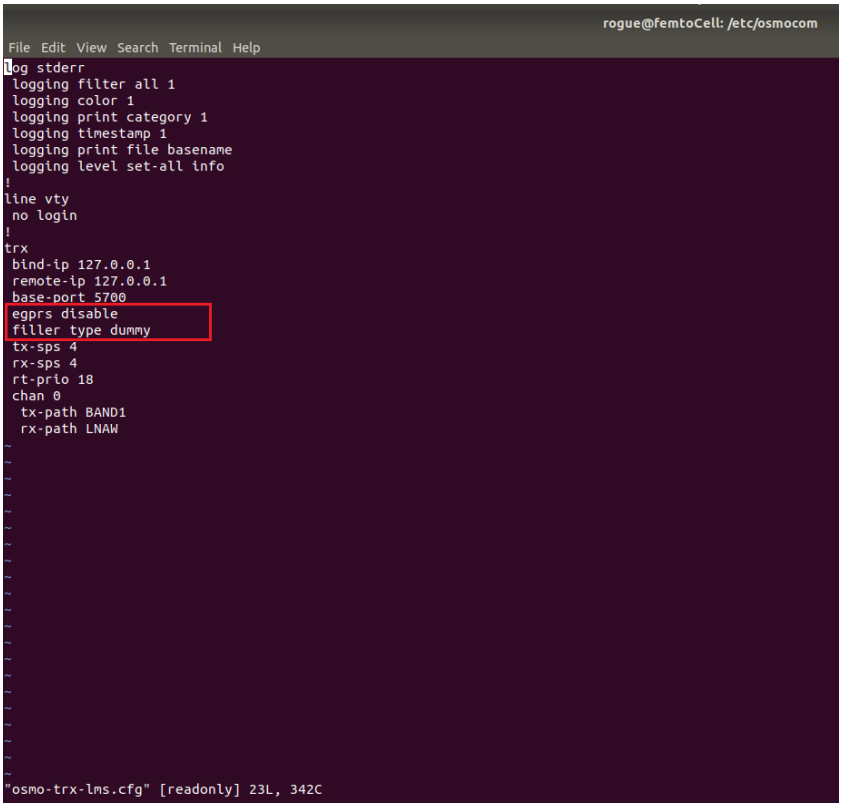

Figure 3.2.2 - osmo-trx-lms.cfg configuration

- The VTY filler type setting sends a Dummy Burst on C0 (TRX0) and empty burst on other channels when there is nothing to send. This setting is the key for work as the new radiomodem of OpenBTS.[12]
- The VTY egprs setting is set to disable EGPRS (8-PSK demodulation) support.[12]

At this point the integration of LimeSDR mini with OpenBTS has been finished. The last step is to test if everything works as we except.

In order to initialize the rogue femtocell we need to execute on different terminals the following commands with the below order:

1. In Terminal 1 we execute: $ sudo systemctl start osmo-trx-lms
2. In Terminal 2 we execute: $ sudo /usr/local/sbin/smqueue
3. In Terminal 3 we execute: $ sudo /usr/local/sbin/sipauthserve
4. In Terminal 4 we execute: $ sudo /usr/sbin/asterisk –vvvv
5. In Terminal 5 we execute: $ cd /OpenBTS ; sudo ./OpenBTS
6. In Terminal 6 we execute: $ cd /OpenBTS ; sudo ./OpenBTSCLI

As we can see from the figure below, the integration of LimeSDR mini with OpenBTS is successfully completed.

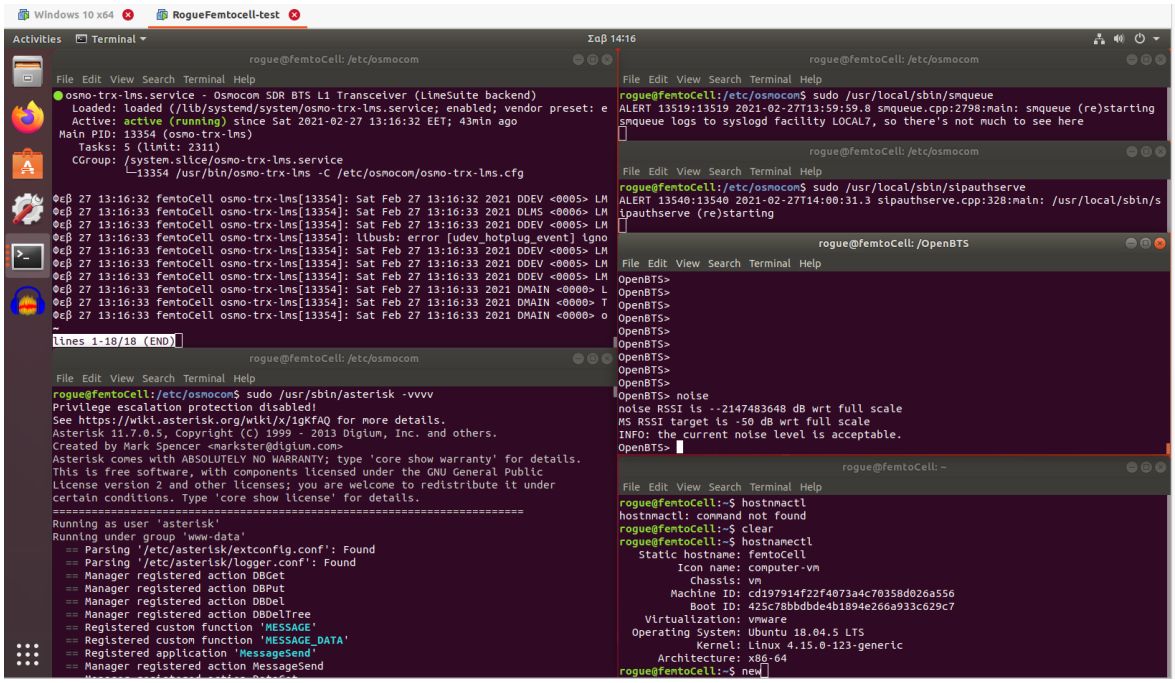

Figure 3.2.3 – Successful Integration

SECTION C – GSM Um ACTIVE ATTACKS

There is a wide category of practical active attacks against the radio interface of the GSM technology in our days. These categories are:

- Communication Interception
- Denial of Service and Service Downgrading
- Location Tracking

The categories that will be covered in this thesis are Communication Interception and Service Downgrading.

Active attacks exploit some of the design weaknesses of the GSM technology itself. An attacker could easily deploy a rogue GSM tower and populate it as a legit service provider tower. With the rogue tower, the attacker could steal the IMSI of the target mobile device. This technique is also called IMSI Catcher. There is no requirement for authentication of the network to the mobile device. Another attack is that the rogue tower acts between the victim's mobile device and the real towers provided by the service provider. This particular attack is also called Man in the Middle Attack. In this thesis we will cover the first attack called IMSI Catcher. The MitM attack is currently under research and will be introduced in future reference.

1.1 IMSI Catcher

The IMSI Catcher is a device that can compromise a GSM network by exploiting the most critical vulnerability, which is that the mobile device connects to the strongest base station signal. Exploiting this vulnerability, a rogue BTS tower which is configured to be identified as a service provider legitimate BTS tower, could deceive the mobile station with the strongest signal and, since the rogue BTS has full control over communication protocols, the mobile device can be maliciously manipulated. For example, it can be instructed in order not to use traffic encryption (A5/0), or to provide critical information like IMSI, or be deceived by impersonating an attacker who is somebody using his MSISDN.

In our LAB environment we will be using the rogue BTS that we have already deployed in the SECTION B. To clarify, in our LAB environment the victim has the capability to communicate with anyone inside our rogue network and to perform outgoing calls to the outside world. The victim cannot receive inbound calls from the outside world.

As a prerequisite for our following attacks are:

- ✓ to be able to associate the MSISDN with the IMSI of the victim.
- ✓ To have already identified the service provider of the victim

The above prerequisites are not practically covered in our thesis.

Taking into account the above prerequisites we assume that the target is the IT Manager of a Bank, his MSISDN is +306947075143 and his service provider is Vodafone. In regards with this information, we move close to the victim and deploy our rogue femtocell with the below settings:

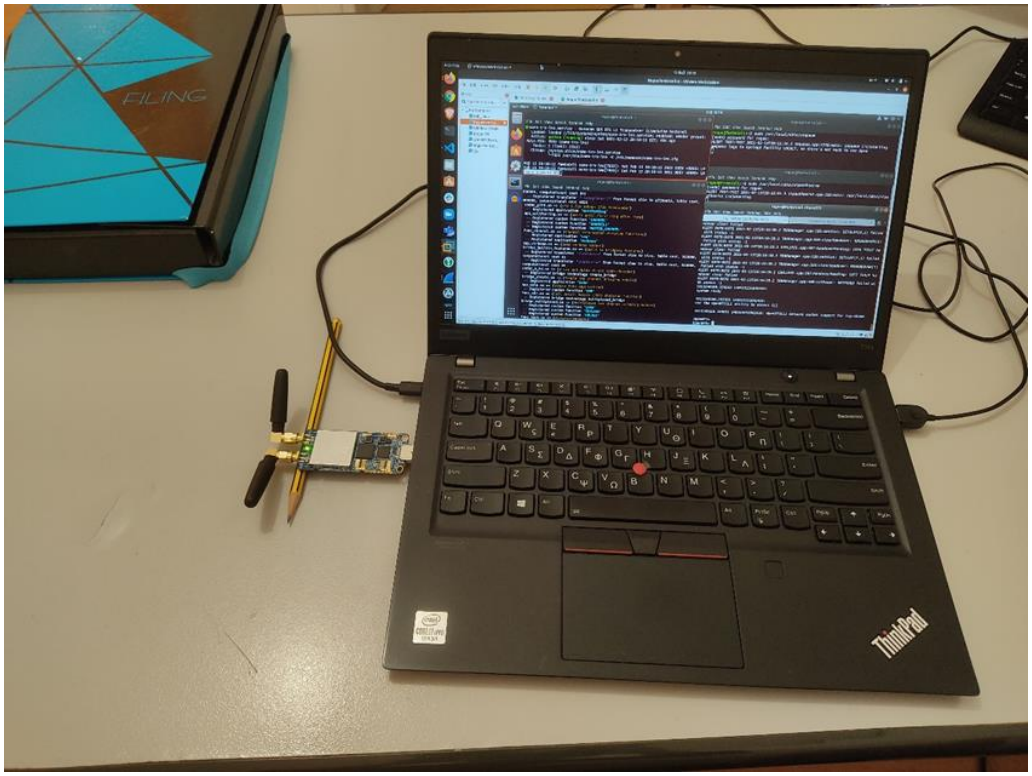

Figure 1.1.1 – Deploying Femtocell

We first change the MCC to 202 which is for Greece, and the MNC to 05 which represents Vodafone Greece as the service provider.

```
GSM.Identity.CI 10      [default]
GSM.Identity.LAC 37
GSM.Identity.MCC 202
GSM.Identity.MNC 05
GSM.Identity.ShortName femtoCell
```

Figure 1.1.2 – Spoofing Vodafone

For a more successful Location Update Request, we change the LAC value from 27 to 37. This process should be done due to the fact that in the victim's area the LAC for Vodafone systems is 27. In order to deceive the victim's mobile device that it has changed LAC, we give another value to it.

From the cellmapper.net we can find out in which LAC the victim is roaming at, as shown in the figure below:

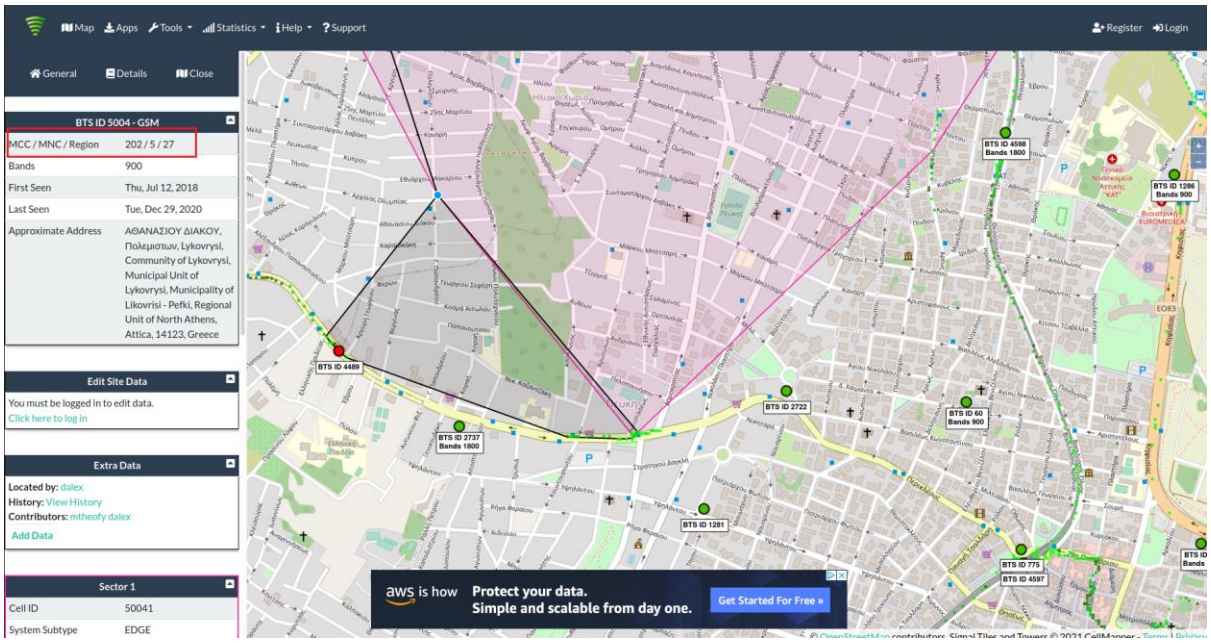

Figure 1.1.3 – Finding the LAC

Then, we leave the radio band as the default:

```
GSM.Radio.Band 900      [default]
GSM.Radio.C0 82
```

Figure 1.1.4 – GSM Band

The next step is to configure OpenBTS to allow any mobile device with an IMSI to register to our rogue femtocell, even if it is not provisioned:

```
Control.LUR.OpenRegistration .*
```

Figure 1.1.5 – Open Registration

With these simple settings configured in our rogue femtocell, we are ready to perform the first attacks.

1.1.1 Attack Scenario 1: LTE Physical Layer Jamming Attack

In order to force a victim's mobile device to perform a Location Update Request and to be downgraded to GSM, we need to jam the LTE downlink channel of Vodafone's eNB to which the victim is connected.

In LTE networks if the signal strength is above a certain sufficient threshold, the mobile station will not scan for other towers to connect to in order to save power. Additionally, the mobile stations when they are connected to LTE networks, they keep track of the nearest neighbors (EARCN) list that is broadcast from the tower that they are connected to. If the connection between the MS and the tower it is connected to, is lost, it will try to connect to one of the available towers that were advertised in the nearest neighbors list first, before doing a full cell reselection scan of the available LTE bands for other eligible cell towers. [24]

In order this attack to be successful, EARFCN or UARFCN must be available. If EARCN or UARFCN is available, we can use this information respectively and we can jam the 4G band by forcing the Mobile Station to perform cell reselection and camp to the strongest GSM BTS available which is our rogueBTS.

The UTRA absolute radio frequency channel number (UARFCN) and the EUTRA absolute radio frequency number (EARFCN) indicate the 3G and 4G neighbour cells. [36]

Again, by using cellmapper.net we confirm that the LTE downlink frequency to which the victim's mobile device is connected and, in our case, it is 806MHz:

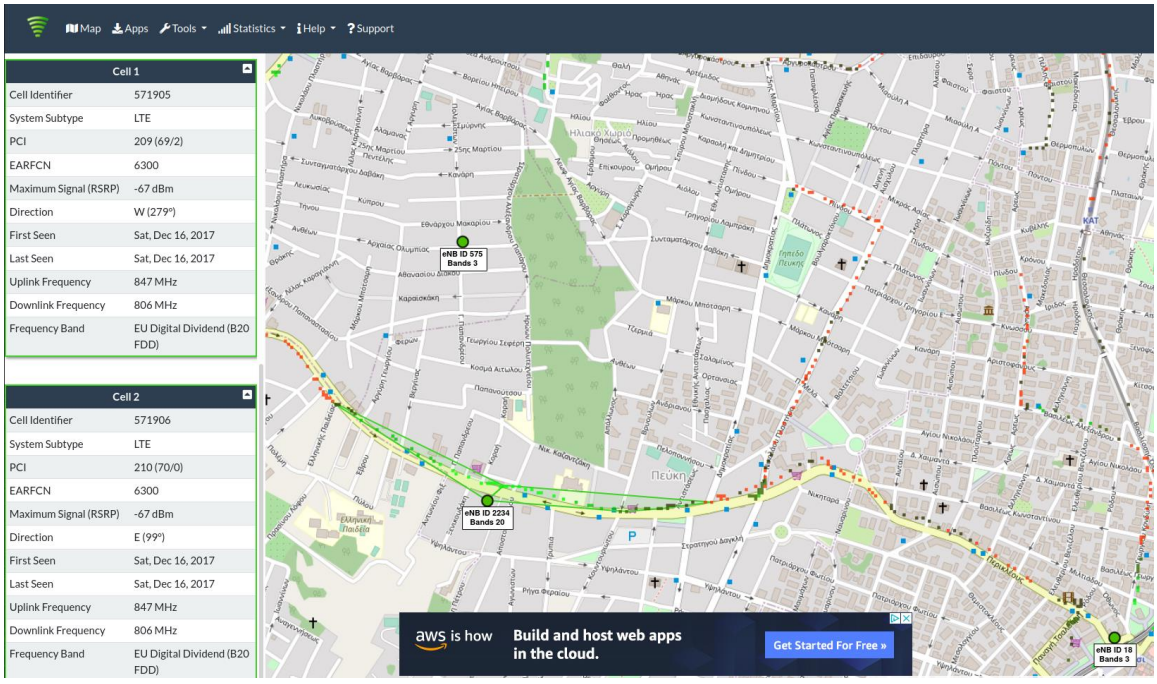

Figure 1.1.6 – Identifying LTE Band

Taking into account this information, we generate a Gaussian distribution signal (White Noise) in GNU Radio Companion, using the same frequency as the Vodafone's eNB channel, which is 806MHz. By

distributing white noise in 806 MHz frequency, the signal from Vodafone's eNB will be jammed and the victim's mobile device is forced to connect to our rogue femtocell network:

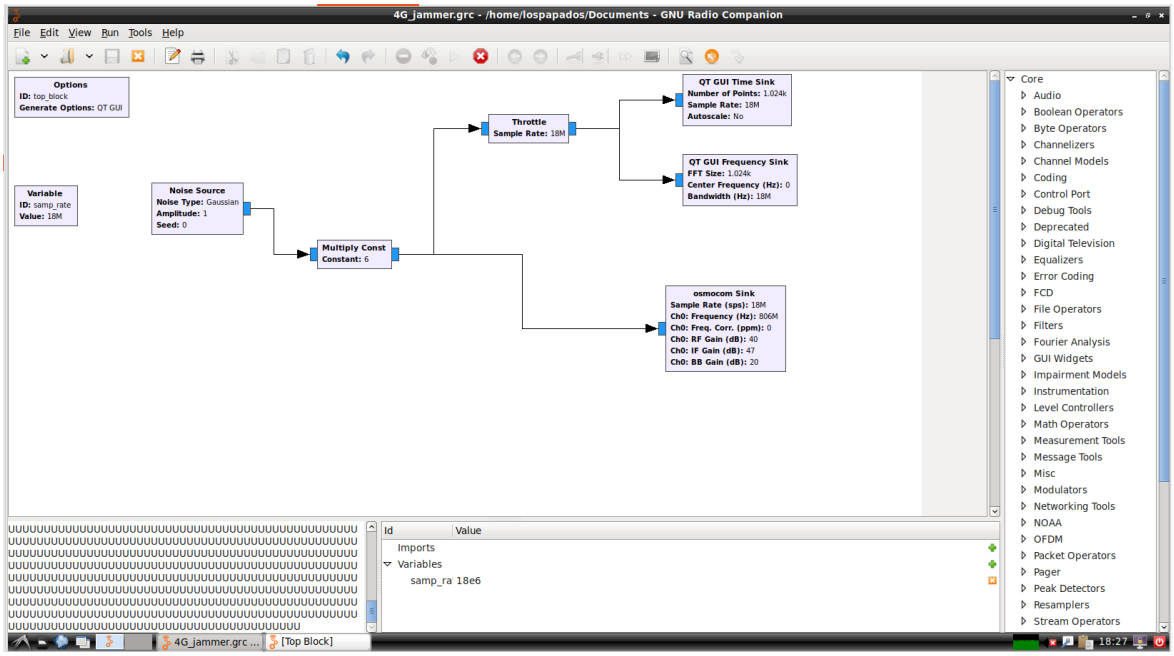

Figure 1.1.7 – Creating LTE Jammer on GNU Radio

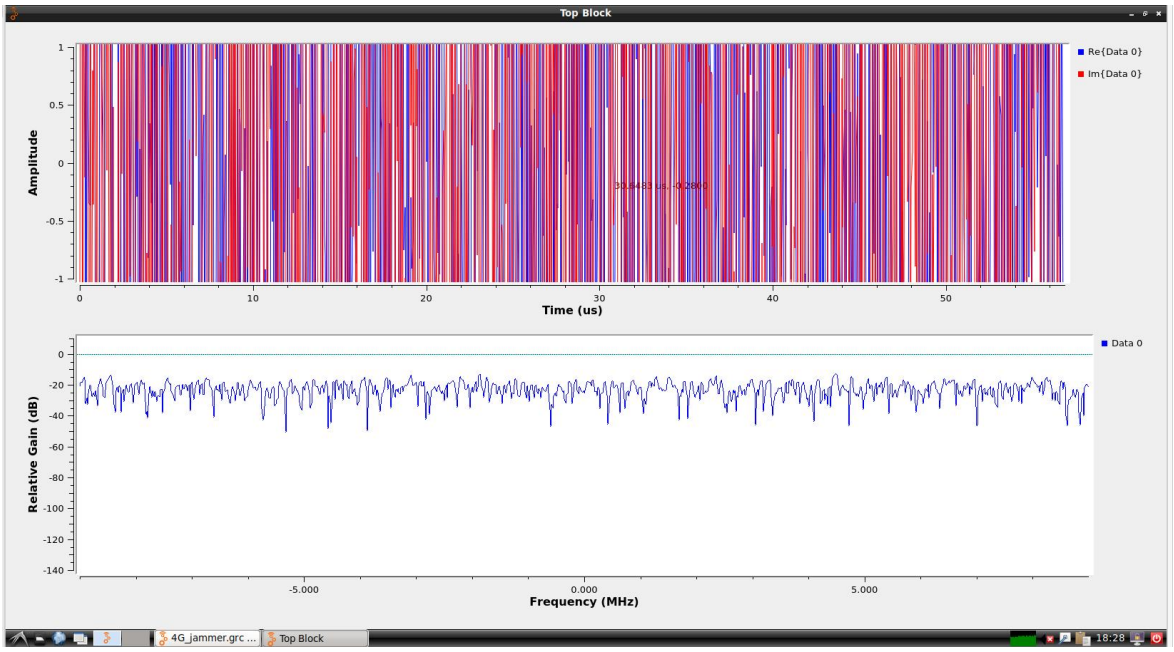

Figure 1.1.8 – Performing LTE Jam in Air Interface

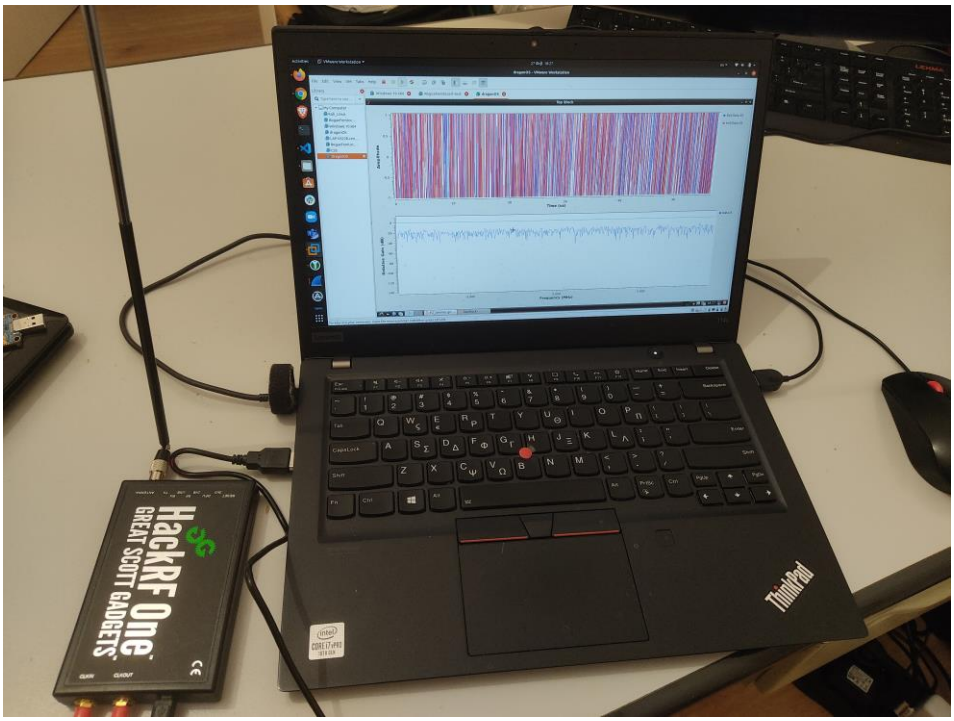

Figure 1.1.9 –Live Performance of LTE signal jamming

Our victim's mobile device is forced to perform cell reselection and automatically camp to our strong signal rogue femtocell as shown in the below figure

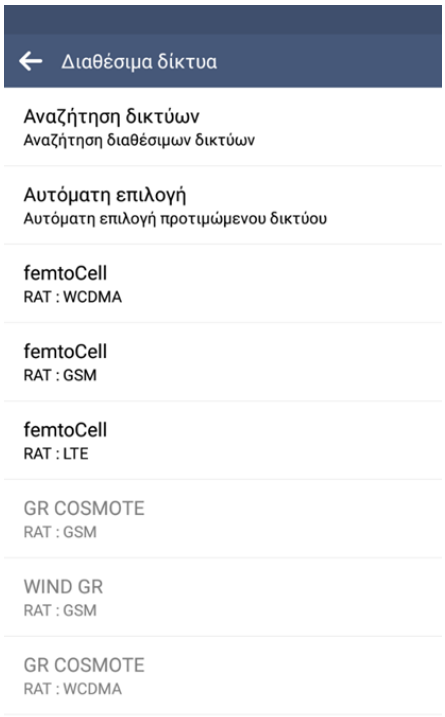

Figure 1.1.10 – Registration to Rogue Network

From the rogue femtocell's perspective, we can confirm the successful LUR of the mobile device with the tmsis command in OpenBTS:

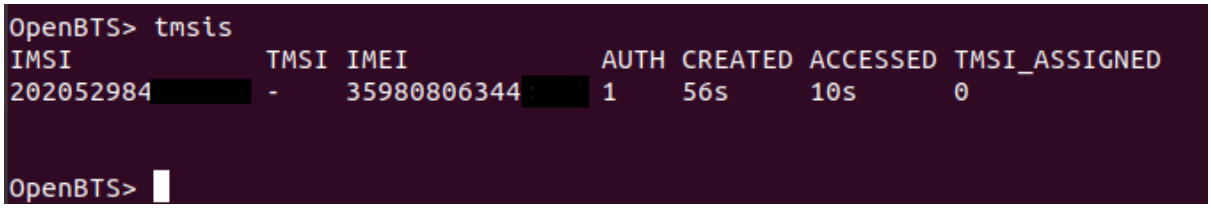

Figure 1.1.11 – Target's IMSI after Registration

Now that we 've got our victim to our rogue network, we can proceed with the following attack scenario which is the SMS impersonation.

1.1.2 Attack Scenario 2: SMS Impersonation

In the second attack scenario we will send a spoofed SMS message to the IT manager of the Bank by pretending that we are a security engineer from OTE. This attack will be executed from the OpenBTS by specifying the victim's IMSI, the source number that the message should appear to have originated from, and the message body itself. In OpenBTS cli we execute the following command:

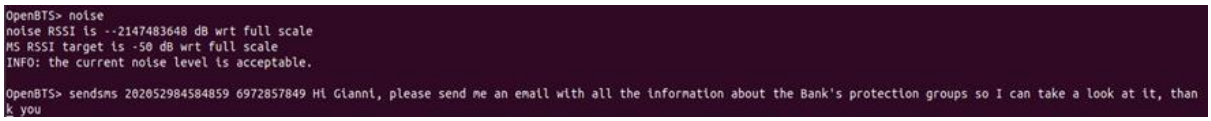

Figure 1.1.12 – Spoofing SMS

In our victim's mobile device, an sms message will appeared just like any other normal sms message, as shown in the below figure:

Figure 1.1.13 – Spoofed SMS delivered to target

The phishing sms message has been delivered to the victim's mobile device and the attack is successfully completed.

1.1.3 Attack Scenario 3: Call Interception

In the third attack scenario, we will try to intercept the call when our victim tries to make an outgoing call on his division where he works. Before we begin with the attack, we need the Asterisk server, which is used as an MSC component, to be configured in order for a client to be able to perform outbound calls. In order to achieve that, we first need to register Asterisk to the ims server of OTE. This could be done from the sip.con configuration file of Astersik [15]:

$ cd /etc/asterisk; sudo vi sip.conf

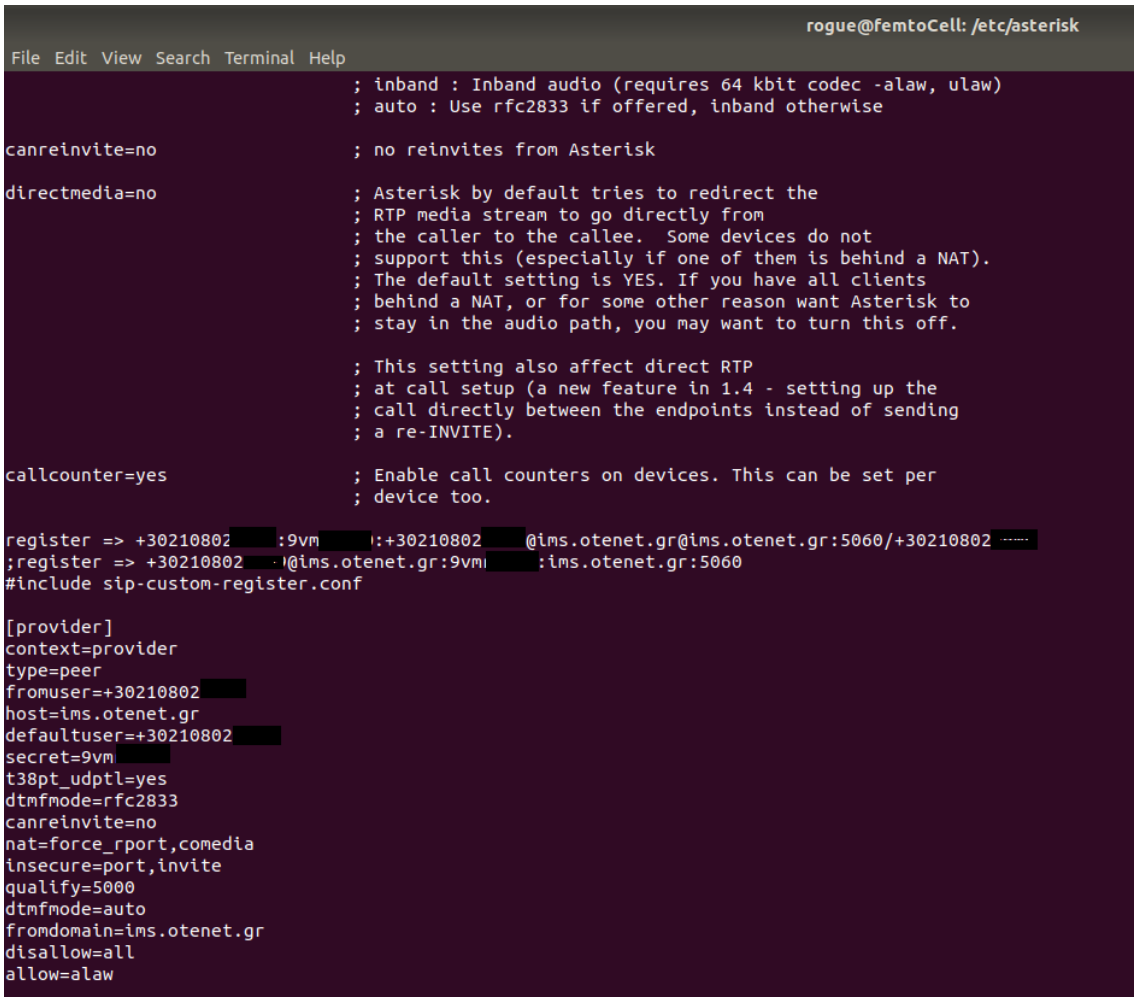

Figure 1.1.14 Registering Asterisk to SIP Provider

As we can see from the above configuration on the sip.conf file, the first step is the registration of the Asterisk to the OTE ims server by providing as username the fixed line number and the secret in the register setting.

The second step, which can also be seen from the image above, is to create a peer for our SIP provider (OTE). This step allows asterisk to route outgoing calls to outside world.

Now, we configure the peer for the victim's mobile device in order to be able to perform outbound calls.

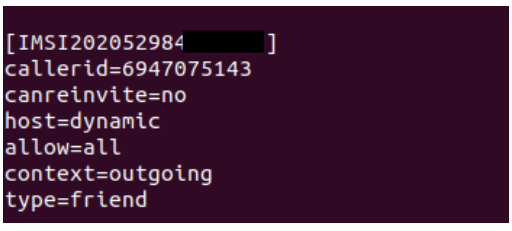

Figure 1.1.15 – Creating peer for our target

After the configuration is made, we run the following command from the cli of Asterisk in order for the configuration changes to be applied:

asterisk*CLI> sip reload

The last step is to configure the dial plan of Asterisk. The dial plan is a context configuration file in Asterisk, where it determines how to handle the traffic for every phone. So, we configure the context 'outgoing' for the victim's mobile device, in order Asterisk to handle outgoing calls of victim's device, and the MixMonitor tool that we will discuss later in extension.conf file:

```
[outgoing]
exten => _X.,1,Dial(SIP/provider/${EXTEN})

exten => 6947075143,1,Answer
exten => 6947075143,n,MixMonitor(6947075143.wav)
exten => 6947075143,n,Dial(SIP/IMSI202052984584859)
```

Figure 1.1.16 – Configuring Dialplan and MixMonitor

And lastly, on asterisk cli we have to reload the dial plan in order the setting to be applied:

asterisk*CLI> dialplan reload

Now we can proceed with the call interception procedure, which can be performed in two ways from inside the rogue femtocell.

In the first way that we will do this, we will use Wireshark to catch the RTP packets from the loopback interface at the time the call is progress:

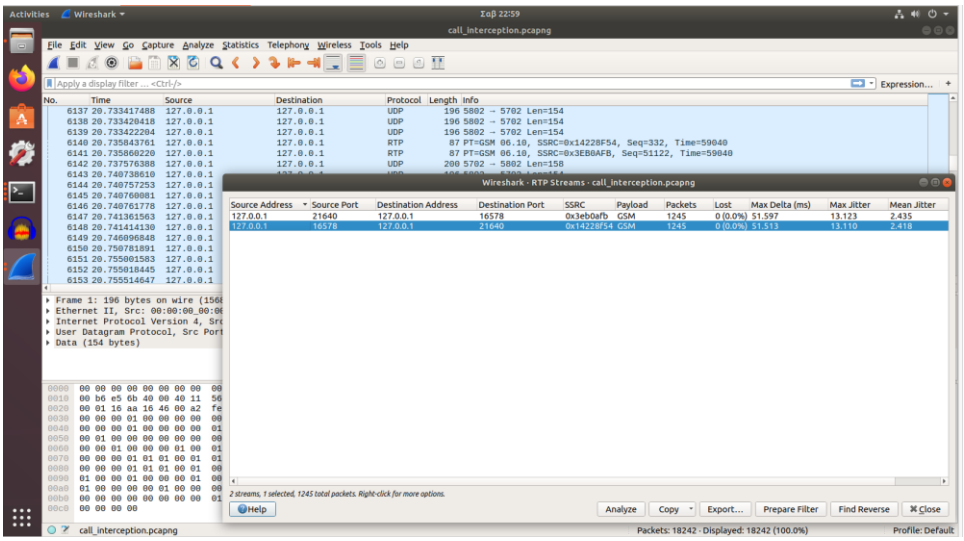

Figure 1.1.17 – RTP Streams

Then we should go to the Telephony tab on Wireshark and navigate to RTP -> RTP Streams tab as shown in the above figure.

We select one of the two channels and we click on the Analyze button in order to export it as a .raw file.

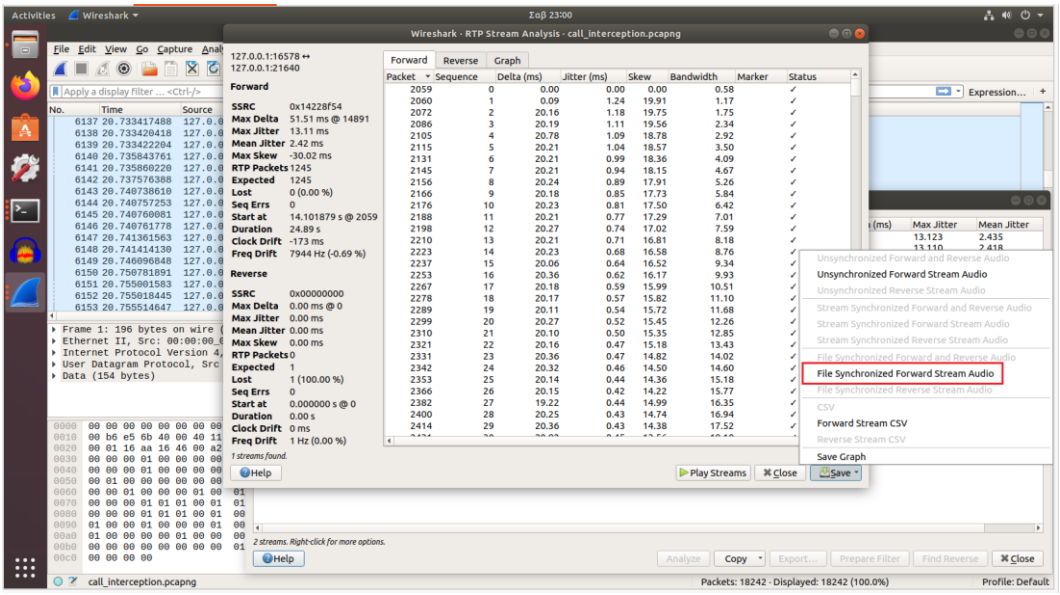

Figure 1.1.18 – Exporting channels in .raw format

In this moment we need to clarify that every channel represents the direction of the voice traffic. For example, the first channel contains the RTP packets from the caller to the callee and the second channel contains the RTP packets from the callee to the caller. So, in order to listen both directions of the call we need to export both channels as .raw files and pass them to the Audacity tool and play back both channels as shown in the above figure:

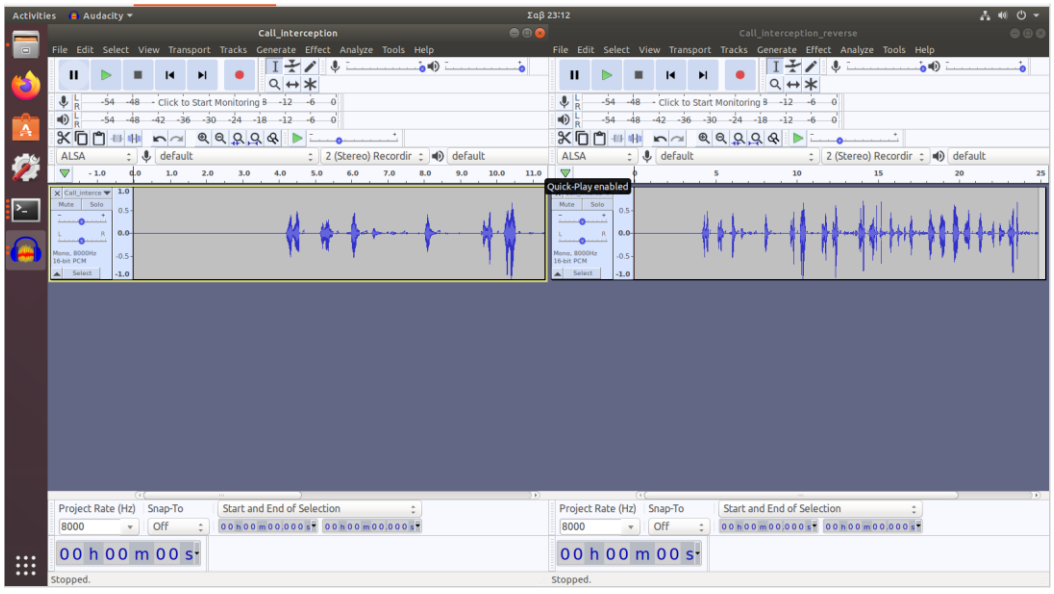

Figure 1.1.19 – Playback channels

The second way is to use an Asterisk built-in tool called MixMonitor. This application is similar to the Monitor application with the only difference that it is designed to record 1 audio and mix both

channels natively as the call is in progress. We have already configured it in the extensions.conf file on the outgoing dialplan.

We can find the .WAV file in the below path of the Ubuntu system and playback the file again with Audacity tool:

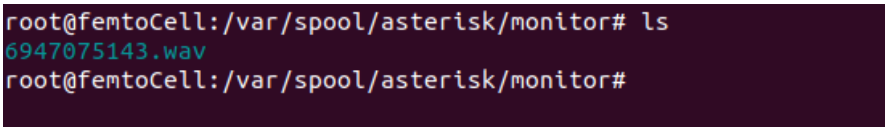

Figure 1.1.20 – MixMonitor wav file

2 Stingrays - Cell Site Simulator attacks

[38] In May 2021, many protesters from every corner of USA have risen against police brutality in order to support the Black Lives Matter movement, some of them have noticed unidentified helicopters hovering above them, which apparently they were conducting surveillance on protesters.

At the end of May, a press document has been released from the Justice Department and it has revealed that the Drug Enforcement Agency and U.S. Marshals Service were asked by the Justice Department to provide unspecified support to law enforcement during protests. A couple of days later a memo received from BuzzFeed News agency provided detailed information on the subject. More specifically it revealed that a few days after the protests took place in some of the cities in USA, DEA had been authorized by the Ministry of Justice to secretly spy on Black Lives Matter protesters on behalf of law enforcement authorities.

Although the press document and the memo did not say what kind of support and surveillance would take place, it is possible that the two agencies were asked to assist the police for a specific reason. Both the DEA and the Marshals have aircrafts equipped with so-called stingrays or dirtboxes, state of the art appliances capable of tracking cell phones or, depending on how they are configured, collecting massive data and communications from cell phones.

[37] Another similar incident took place in Greece on March 2021, as two young men went together to Nea Smyrni in order to participate in the mobilization against police brutality. The one of them a 22-year-old is said to belong to an anarchist group, according to police sources, and in mid-February he was arrested and charged with vandalizing the political office of Deputy Foreign Minister Miltiadis Varvitsiotis. The other person is a 23-year-old Greek-Iranian, organized fan of a football team. At that night he faced charges of attempting murder and four other crimes as the demonstrations turned violent.

According to the facts, the Counter-Terrorism Department was monitoring the anarchist's mobile phone, invoking national security reasons. So, while the riot was in progress, the police recorded his telephone conversation with his 23-year-old friend, confessing that they might have killed a cop in the street riots.

Stingrays which are part of the broader category of Cell Site Simulators (CSSs) have been utilised both in the air and on the ground for a long time. They collect data from any mobile station that is located near them and this data can be used to identify people, track their movements or hijack their conversations.

Although law enforcement has been using this technology since the 1990s, the general public has only learned about it in the last 10 years and much about their capabilities still remain unknown, as law enforcement and companies that manufacture those appliances spend so much effort to keep details secret. Stingray is commonly used to target drug dealers and other criminal investigations, but many activists also believe the devices have been used during protests for example against Black Lives Matter protesters. The Department of Justice requires federal agents to obtain a warrant for possible use of the technology in criminal matters, but there is a gap for national security.

2.1 What is a Stingray appliance?

[38] Stingray is the generic name for an electronic surveillance tool that simulates a cell phone tower in order to force mobile phones and other devices to connect to it instead of a legitimate cell phone tower. In this way, the phone or other devices reveals information about itself and its user to the attacker as well as it is able to perform Man in the Middle attacks. Other common names for the tool are "Cell Site Simulator CCS" and "IMSI-Catcher".

The Stingray name comes from the trademark of a specific Cell Site Simulator model manufactured by Harris Corporation. This particular appliance model is a briefcase-sized device that can be used by a vehicle while connected to a cigarette lighter. Harris also makes products such as the Harpoon, a signal booster that makes the StingRay more powerful, and the KingFish, a smaller handheld device that acts as a stingray and can be used by a law enforcement officer while walking outside a vehicle. There are quite a lot other companies that make variations of the stingray with different capabilities.

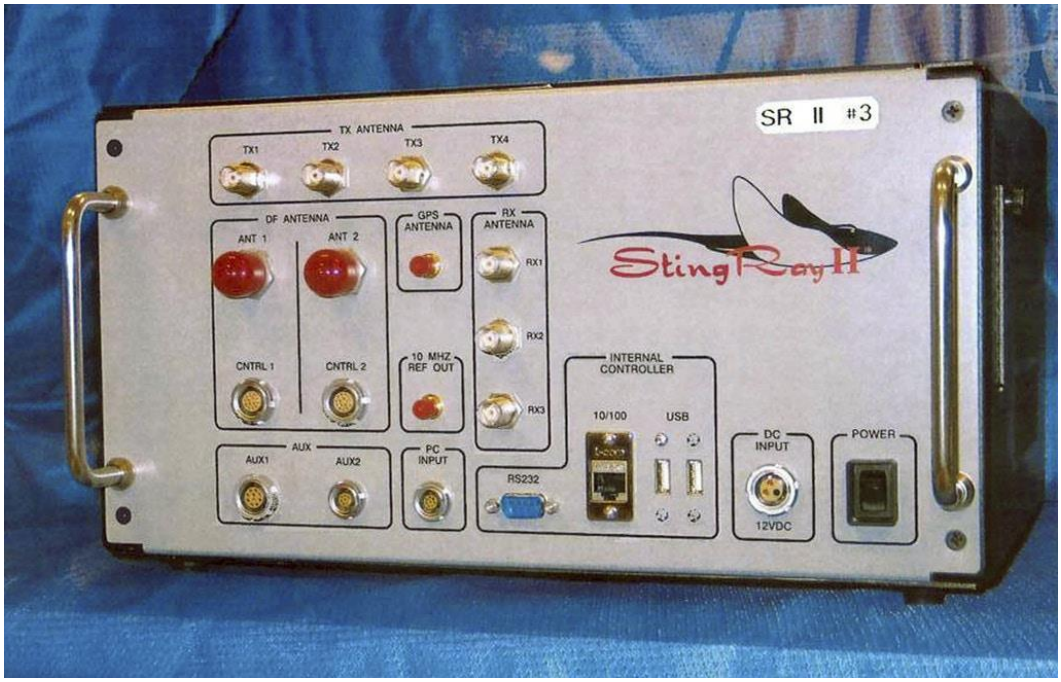

Figure 2.1 - Harris StingRay II cellular site simulator used for surveillance purposes [38]

2.2 Cell Site Simulator type of attacks [51]

There are three main categories of attacks that from the academic perspective we know that Cell Site Simulators are able to conduct:

1. Communication Interception and service downgrading
2. Denial of Service
3. Location Tracking

In this Thesis we will cover Communication **Interception Attack**. Other common name for this attack is Man in the Middle Attack.

2.3 Communication Interception and service downgrading [51]

As we have already cover in this Thesis a Man in the Middle attack in the communication between a mobile station and a legitimate cell tower can be performed only in GSM. When it comes to newer technologies like 3G or 4G, we have to downgrade the technology to 2G/GSM in order to be able to perform the communication interception.

So, GSM is the right technology to perform communication interception because:

- ✓ Encryption over GSM is not required
- ✓ GSM algorithms can be broken and during a real time communication.

2.3.1 Man in the Middle Attack

A man-in-the-middle attack is a type of cyberattack in which an attacker eavesdrops on a conversation between two targets. The attacker may try to "listen" to a conversation between two people, two systems, or a person and a system.

In this Thesis we will try to perform an active attack in order to intercept a victim's mobile station. The rogue Cell Site Simulator system will be placed between the mobile station and the legitimate base station to be able to perform the aforementioned attack.

There are two fundamental steps in order to perform a Man in the Middle attack:

1. Authentication Spoofing. The rogue CSS have to persuade the Network that its actually the targeted mobile station.
2. Reject any encryption the Network tries to set, or negotiate for a weak algorithm or even try to break it.

2.3.2 Spoofing Authentication

Picking up from Section 1.1 where the IMSI Catcher rogue femtocell has already obtained the victim's mobile station IMSI:

1. The CSS system performs a Location Update Request to a legitimate Base Station.
2. In response to the Location Update Request, the Cell Network asks the CSS system to identify itself using an Identity Request. The CSS responds using the stolen IMSI.
3. At this point the BTS responds with a cryptographic challenge that requires the secret key Ki (stored on the SIM card) to solve. Since the CSS doesn't have access to Ki , it passes it onto the victim's mobile station to solve. The victim's mobile station solves the challenge, passes it to the CSS, who then passes it back to the Cell Network.
4. After this, the Cell Network accepts the connection between it and the CSS as being authenticated.

2.3.3 GSM encryption Bypassing

As it is already mentioned the encryption algorithms that have been used by GSM technology are A5/1, A5/2, A5/3 and A5/0 indicates that no encryption is being used.

In the following scenario let's say the Cell Network addresses that it wants to communicate using encryption, the CSS has the ability to respond that it doesn't have encryption capabilities and defaults to A5/0. The CSS has now completed the MitM attack and can read the plaintext messages being sent between the victim's mobile station and the legitimate network.

The good thing about this, from the attacker's side, is that the mobile station does not notify their owners that no encryption mode has been utilised for communication with the Network.

According to the GSM specifications, mobile stations are supposed to notify their owners when encryption is disabled, and in some countries it is enabled. However, this caused a lot of confusion because people would travel with their mobile stations to places where BTS systems have different configurations. For example, in some countries Cell Network encryption is banned and as a result it would cause multiple times that a warning encryption is disabled. Moreover, due to BTS misconfiguration in most of the countries, the pop-up notification alert appeared lot, as a result, to it has been disabled.

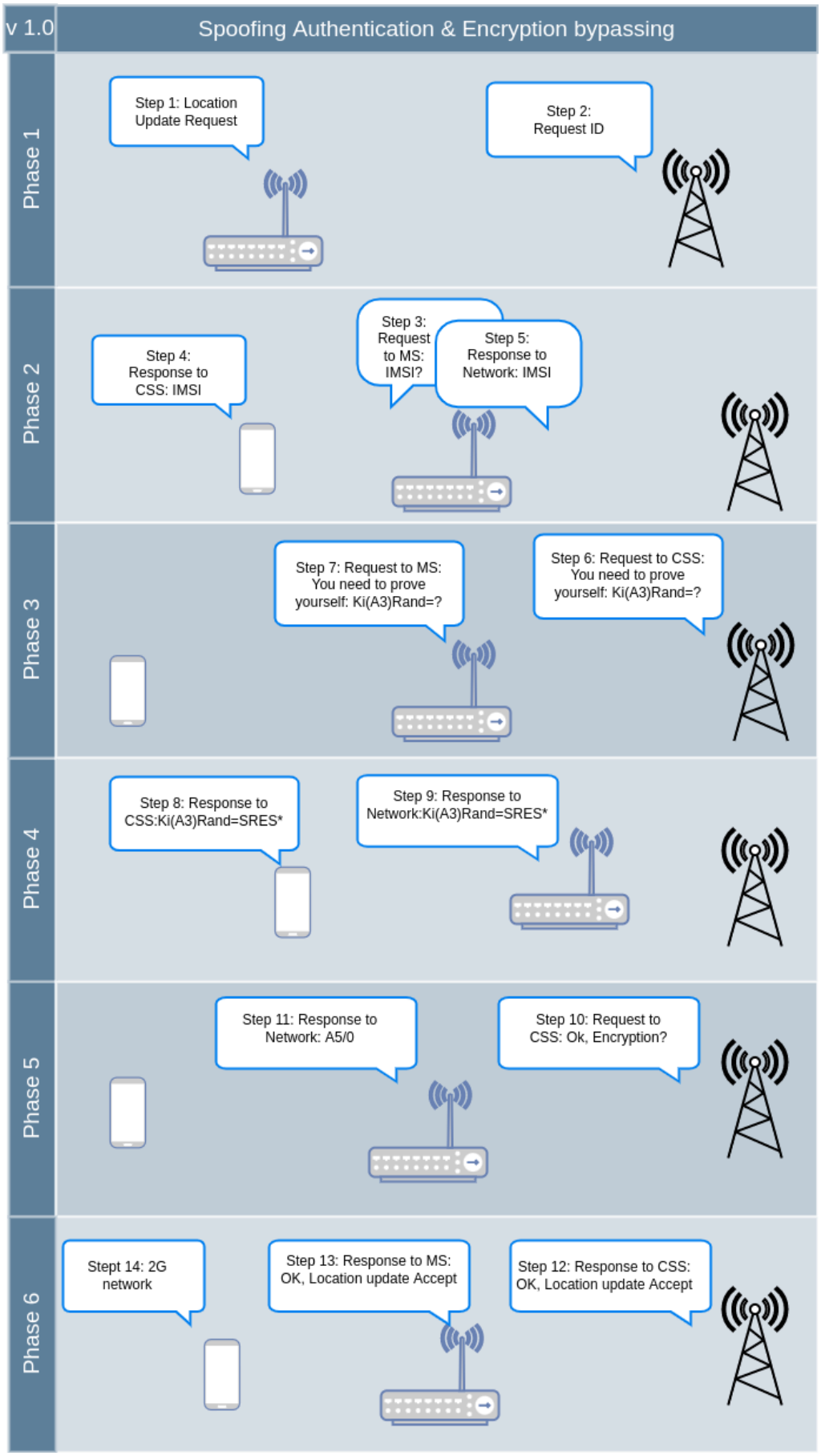

Figure 2.2 – GSM Authentication Spoofing

2.4 Defining the Cell Site Simulator

In this section we will setup the lab environment. By manipulating malicious the Cell Site Simulator we will perform a Man in the Middle attack later in this thesis.

In order the Cell Site Simulator to be able to interact with a legitimate BTS and the Victim's Mobile Station at the same time, we need to create two different interfaces and a 'forwarder' component that will link these two interfaces. On account of this we will setup:

- ✓ The first interface (Base Station Subdomain) that will act as a legitimate Network (BSS & MSC) and it will communicate with the victim's mobile station.
- ✓ The second interface (Mobile Station Subdomain) that will act as a legitimate subscriber and it will communicate with a legitimate Network (e.g Vodafone)
- ✓ And finally, the link between these two interfaces that will forward the request from one interface to the other.

Below the chart represents the Location Update Establishment flow when our CSS system sits between the victim's mobile station and the legitimate Network:

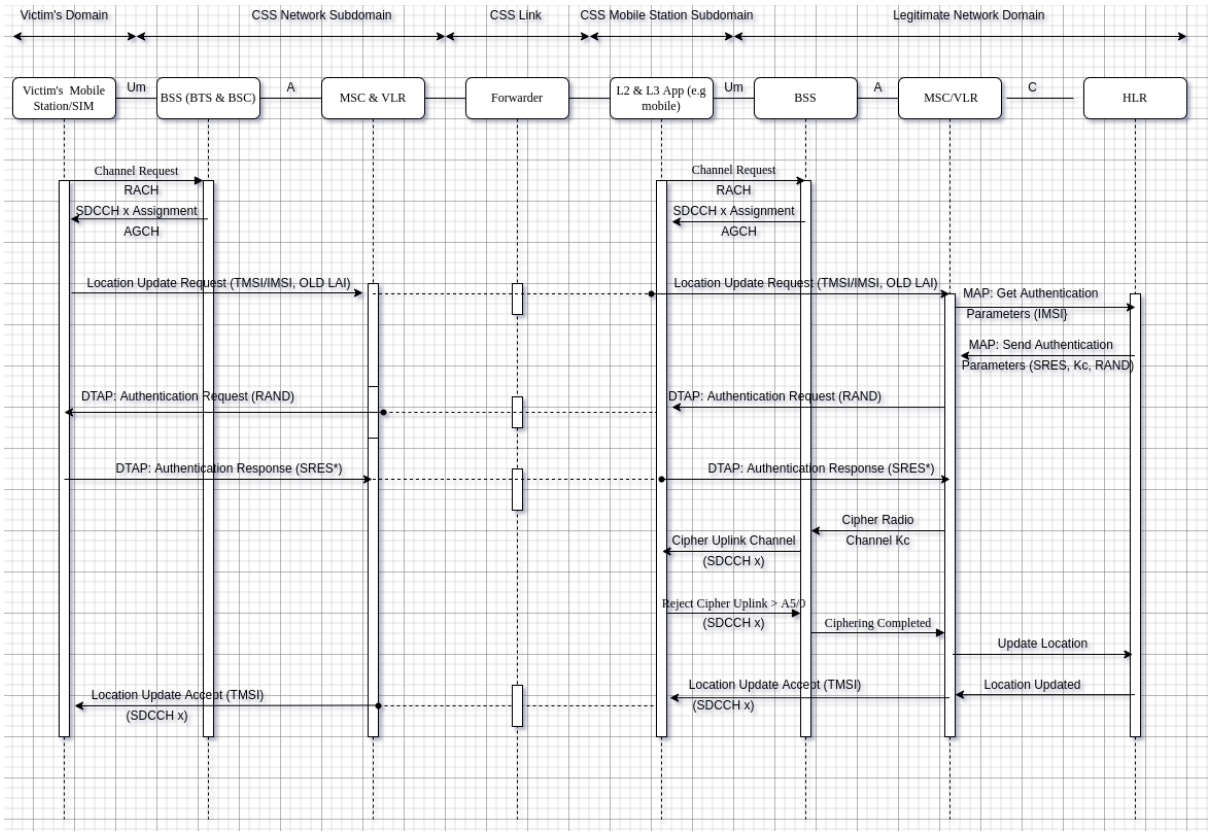

Figure 2.3 – Location Update Establishment - CSS Interception Flow Chart

2.4.1 Osmocom project (osmocom.org)

The Osmocom (Open Source Mobile Communications) project is an open-source project which comprehends software and tools that implements a variety of mobile communication standards, including GSM, DECT, TETRA and others.

The next step is to define the components of the Cell Site Simulator system that will perform the above functionality. In order to perform that we will use components from the Osmocom project and we will deploy them according to the below architecture chart:

As we have already worked with **Limesdr mini** and **osmotrx-lms** in the previous sections, we will introduce the rest of the components of the Osmocom Project that we are going to use for the sake of the Cell Site Simulator system.

2.4.2 OsmoTRX [41]

OsmoTRX is a software-defined radio transceiver that implements the physical layer (Layer 1) of a BTS system consisting of the following 3GPP specifications:

- ✓ TS 05.01: Physical layer on the radio path
- ✓ TS 05.02: Multiplexing and Multiple Access on the Radio Path
- ✓ TS 05.04: Modulation
- ✓ TS 05.10: Radio subsystem synchronization

OsmoTRX is based on the transceiver code from the OpenBTS project, but is configured to work independently regarding integration with other Open source Mobile Communication software and projects, while maintaining backward compatibility with OpenBTS project.

2.4.3 OsmoBTS-trx [40]

OsmoBTS is an Open Source GSM BTS (Base Transceiver Station) which comes with A-bis/IP interface. It implements support for a variety of PHY/Hardware, such as sysmocom sysmoBTS, OCTSDR-2G and LiteCell 1.5, but also general-purpose SDR boards such as USRP or LimeSDR via OsmoTRX.

More specifically, It implements the follwing protocols/interfaces:

- ✓ LAPDm (GSM 04.06)
- ✓ RTP
- ✓ A-bis/IP in IPA multiplex
- ✓ OML (GSM TS 12.21)
- ✓ RSL (GSM TS 08.58)

OsmoBTS is modular and has support for multiple back-ends. A back-end talks to a specific L1/PHY implementation of the respective BTS hardware. Based on this architecture, it should be relatively easy to add a new back-end to support so-far unsupported GSM PHY/L1 and associated hardware. So far OsmoBTS has been integrated with several different L1/PHY and hardware systems. The backend that we will use in our deployment is osmo bts-trx:

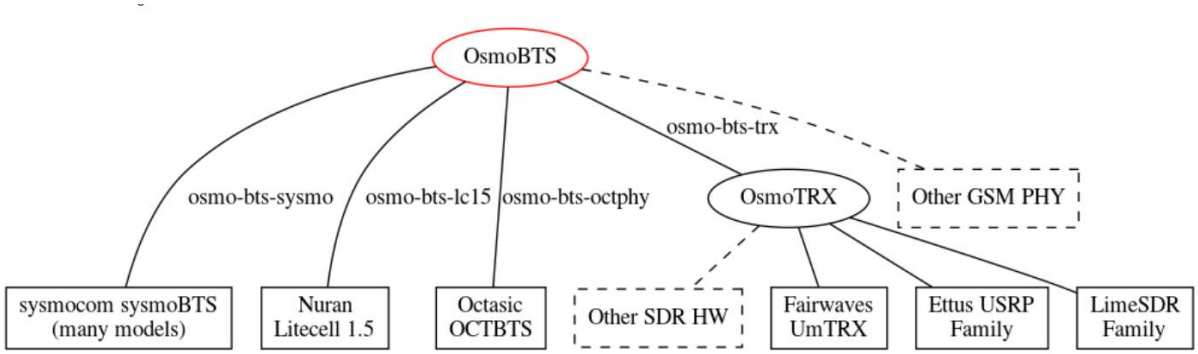

Figure 2.4 – osmoBTS Backends for Hardware support

2.4.3 OsmoBSC [42]

OsmoBSC is an implementation of a GSM BSC (Base Station Controller) and it performs the below functionalities:

- ✓ an A-bis interface towards the BTS systems
- ✓ an A-over-IP (AoIP) interface towards an MSC system (e.g. OsmoMSC).
- ✓ an Media Gateway Control Protocol (MGCP) interface towards a MGW (e.g. OsmoMGW for handling the RTP user plane (voice call codec frames)

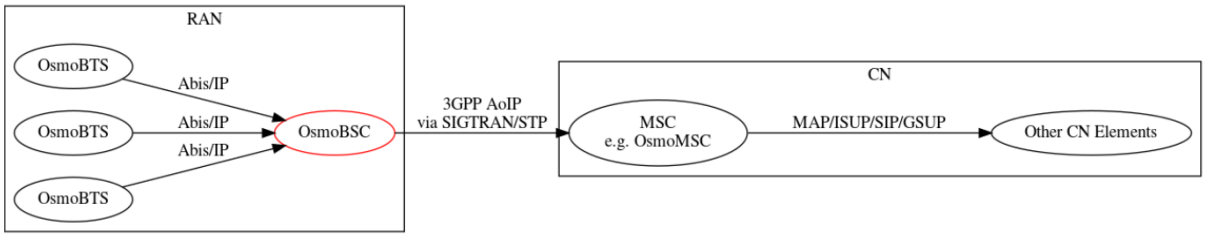

Figure 2.5 - OsmoBSC in the Osmocom architecture

2.4.4 OsmoMSC [43]

OsmoMSC is the Osmocom implementation of a GSM Mobile Switching Center (MSC).

It implements the following interfaces:

- ✓ 3GPP AoIP over M3UA or SUA towards BSC systems (e.g OsmoBSC)
- ✓ 3GPP IuCS over M3UA or SUA towards RNCs or HNBGW systems (e.g OsmoHNBGW),
- ✓ Osmocom GSUP (alternative to SS7/TACP/MAP) towards an HLR system (e.g OsmoHLR)
- ✓ SMPP v3.4 for external SMS entities (minimal SMSC is built-in)
- ✓ MGCP for controlling an external Media Gateway systems (e.g OsmoMGW)
- ✓ MNCC for external call-control handlers, such as osmo-sip-connector for SIP trunks
- ✓ Osmocom VTY interface for configuration + introspection
- ✓ Osmocom CTRL interface for programmatic access to internal state/configuration

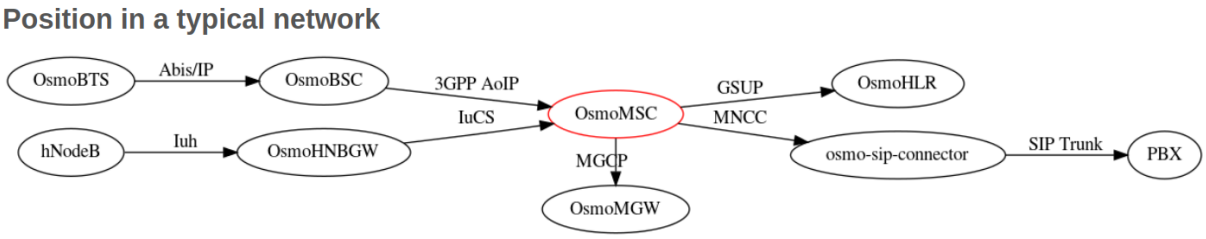

Figure 2.6 - OsmoMSC in the Osmocom architecture

2.4.5 OsmocomBB project [44]

OsmocomBB is a Free Open Source GSM Baseband software implementation. More specifically, It is a free firmware for the baseband processor of mobile phones which handles the encoding and radio communication of both voice and data. OsmocomBB is the only existing free implementation of baseband firmware and It intends to completely replace the need for a proprietary GSM baseband software, such as:

- ✓ drivers for the GSM analog and digital baseband
- ✓ drivers for (integrated and external) peripherals
- ✓ the GSM phone-side protocol stack, from layer 1 up to layer 3

By using OsmocomBB on a compatible phone, that we will discuss in the following section, it will be feasible to send or receive phone calls and send or receive SMS messages as well.

2.4.5.1 Hardware [45]

There is variety of devices and hardware that are supported by OsmocomBB project or currently work-in-progress and are listed in the following link: https://osmocom.org/projects/baseband/wiki#Introduction.

The aforementioned link also provides information for the Calypso based phones that are supported by OsmocomBB.

Nowadays, the supply of Motorola/Compal phone devices is limited, and the quality that is currently available in most cases are repaired (not just refurbished) phones. However, we have managed to acquire one piece of a new Motorola C123. The specifications for Motorola C123 are listed below:

- ✓ GSM 900 / GSM 1800 dual-band
- ✓ no GPRS
- ✓ Ti Calypso/Iota/Rita chipset
- ✓ GSM Chipset: DBB: Ti Calypso Baseband, D751749ZHH model (Calypso Lite G2), includes 256kBytes of internal SRAM

Figure 2.7 – Motorola/Compal C123

2.4.5.2 Calypso Digital Baseband [46]

The Calypso Digital Base Band chip is a popular DBB implementation for inexpensive phone devices. The register-level manuals have leaked publicly and are available from cryptome.org.

2.4.5.3 Serial Port [47]

The Calypso phone devices typically all have a serial port @ 3.3V levels on the 2.5mm earphone jack. The cable that we have acquired comes as USB serial cable and can be used for establishing a connection between a PC and the UART in OsmocomBB-compatible phone.

Figure 2.8 – Motorola T191 cable, including a CP2102 based USB-Serial converter.

2.4.6 Osmocon [48]

Osmocon is a console tool which is responsible for interfacing the baseband firmware on the phone device with applications in the host PC. It is also used to download a firmware or bootloader into the phone device through the serial link. When the uploading process of the firmware is completed, it turns into an HDLC mulitplexer/demultiplexer, allowing for multichannel communication with the phone device.

The console is on one such channel and will be redirected to the terminal (stdout) on which osmocom runs.

Various other HDLC channels are accessible by the following unix domain sockets:

- ✓ **/tmp/osmocom_l2** for the L1A_L23_Interface as used by mobile, ccch_scan and other host programs
- ✓ **/tmp/osmocon_loader** for the Bootloader

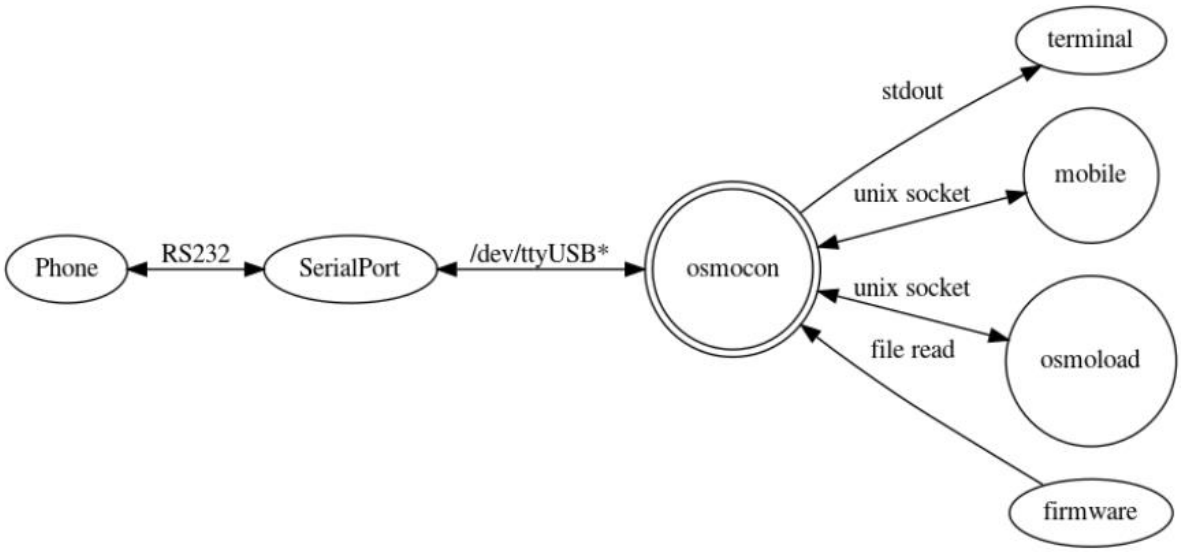

Figure 2.9 – Osmocon tool in OsmocomBB architecture

Generally, the osmocon application is responsible for the firmware loading process and message forwarding between hardwareCalypso based phone and mobile application.

2.4.7 OsmocomBB software stack [49]

In this section we will explain how osmocomBB software stack receives a GSM signal.

1. As a first step in our scenario, an RF signal is received at the antenna of the Motorola C123, and go through the Rita mixer which converts it into analog I/Q baseband. Then it goes through the baseband ADC of the Ioata ABB component. The processed signal is passed to Baseband serial port of the HardwareCalypso DBB:

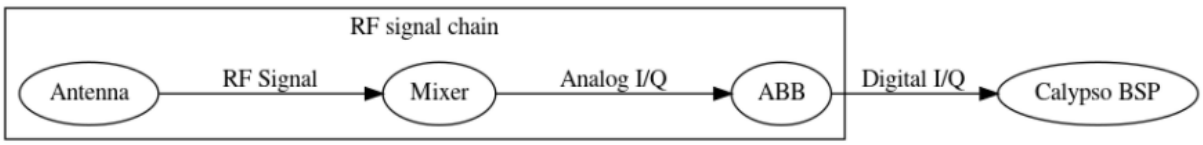

Figure 2.10 – RF signal chain

2. At this point the digital baseband samples that came to the HardwareCalypso DBB, are passed from the BSP to the DSP core, where they are processed, demodulated, deinterleaved and decoded, before being passed onto the ARM core using a shared memory interface. On the ARM core we have the OsmocomBB layer1 that processes the MAC blocks and sends them via L1CTL to the UART:

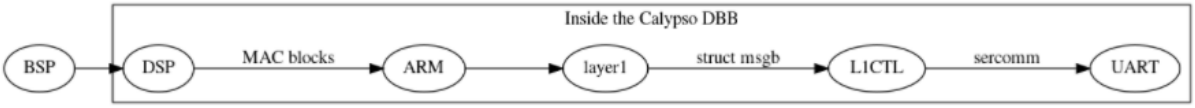

Figure 2.11 –Inside Calypso DBB

3. On the host PC, the L1CTL messages are received through the serial port by the osmocon program, which demultiplexes the different sercomm streams and forwards L1CTL messages via a unix domain socket into a layer23 program which in our deployment is mobile application:

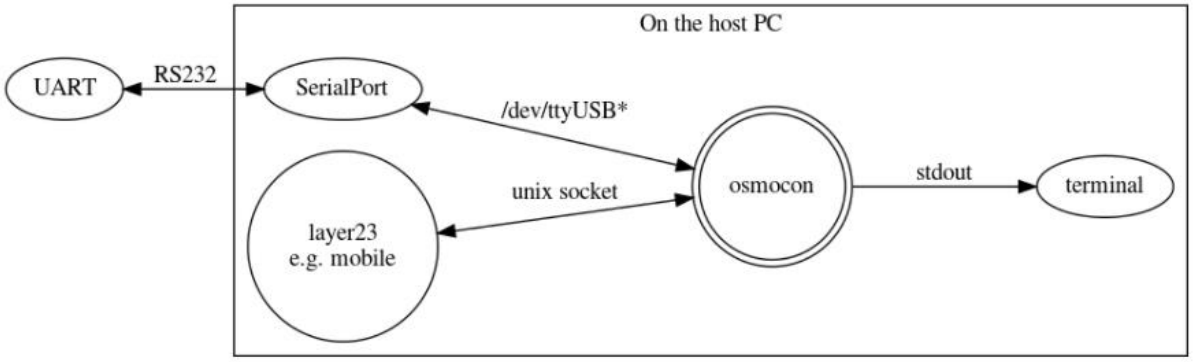

Figure 2.12 – Host PC Software

Host software is software that runs on the host PC, and not inside the phone itself.

The Location of the source and binary code is located on a unix based host at the following path: **src/host/layer23/***

Layer 3 applications implement various functionality based on the combination of GSM Layer 3 and Layer 2 (LAPDm). In our CSS system we will be using **mobile** as a Layer 3 application.

2.4.7.1 mobile [50]

From the OsmocomBB L2&3 software, **mobile** is the most sophisticated application so far. It implements most of the behavior of a regular GSM mobile phone, but it can be extended in many ways with features interesting to the academic world. Some of the features are:

- ✓ perform cell (re)selection according to TS 03.22
- ✓ MM procedures like location updating, authentication, encryption
- ✓ Establish MT and MO voice calls
- ✓ Send and receive SMS
- ✓ Perform supplementary services like USSD or call forwarding
- ✓ Connect it to a PBX

2.4.8 Branches [52]

There are several branches in the project repository. Some project changes are incompatible with each other, so they exist in separate branches. In our deployment we will use master Branch which is the main branch of OsmocomBB. The list of branches can be found in the following link: https://osmocom.org/projects/baseband/wiki/Branches

2.5 Implementing Cell Site Simulator with Osmocom components

As we have defined the functionality and reviewed the Osmocom components of the Cell Site Simulator system we will proceed with connecting those pieces together in a way to reflect the aforementioned functionality. Our lab Cell Site Simulator system will look like the below chart architecture:

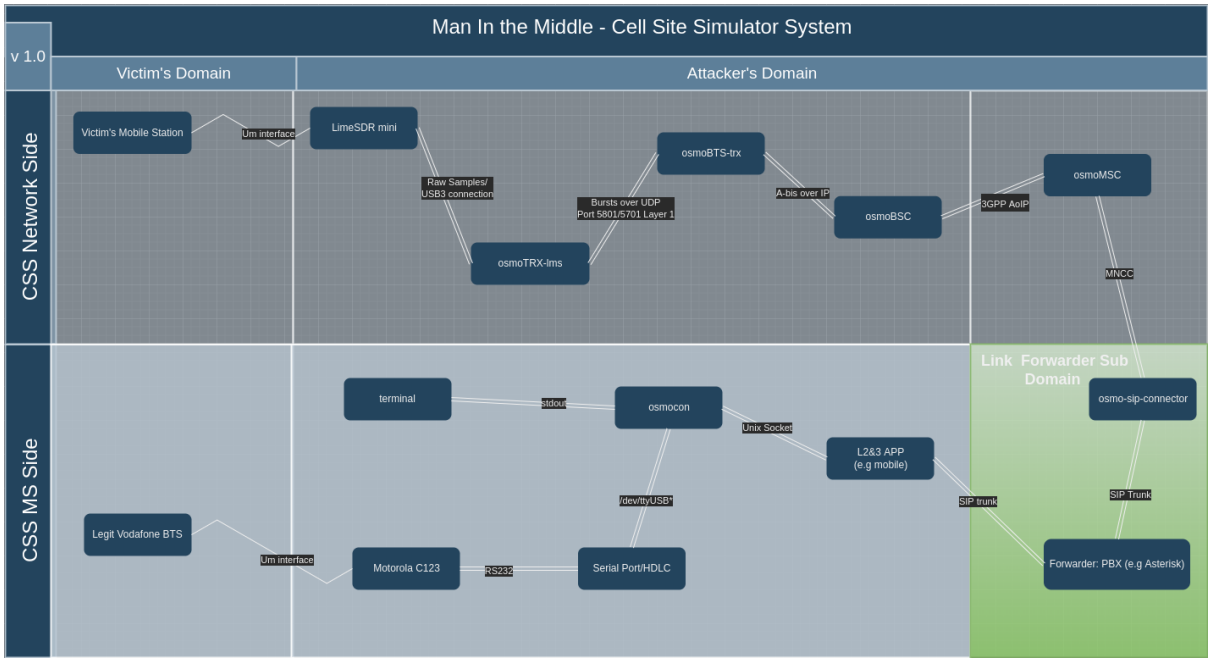

Figure 2.13 – CSS Component Architecture

As we have already covered in the previous sections, our CSS system consists of three sub domains. The first subdomain is the CSS Network side which interacts with the victim's Mobile Station device. The second sub domain is the CSS Mobile Station Side which interacts with the legitimate Network, for example Vodafone's Network. And the third sub domain is the link between the aforementioned two sub domains.

2.5.1 CSS Network Sub Domain

The first phase is to setup our CSS BSS. In order to achieve this, we will use our limeSDR mini to act as our BTS system. In an Ubuntu 18.04 LTS Virtual machine we will install and configure osmo-trx-lms which will take care of the SDR side of the Sub Domain and will serve as a modem. It will send everything it gets on the Um interface to osmo-bts-trx, which must be also installed and configured on the same VM, over UDP as well as everything it receives from osmo-bts-trx over UDP it passes it through the Um Interface. osmo-bts-trx will then setup an Abis over IP connection to our BSC system.

When it is done with deploying the Base Transceiver Station (BTS), it cannot work unless it connects itself to a Base Station Controller (BSC). In practice this means the BSC configures most of the parameters on the BTS and brings each one up onto the air when they are ready. Therefore, in our VM we will install osmoBSC component and configure it accordingly. The last component that we should install on our VM for handling calls to and from subscribers as well as authenticating them, is the osmoMSC component.

2.5.2 CSS Mobile Phone Sub Domain

The second phase is to setup our Mobile Station. It can be performed with the use of OsmocomBB tools. Our Motorola/Compal C123 mobile phone will need to be connected to our VM Ubuntu 18.04 via the Motorola T191 serial/usb cable. After the Motorola C123 is connected to our VM, by executing osmocon we will perform a connection between the baseband firmware on the phone with applications on the host PC. It will allow us for downloading a baseband firmware or bootloader into the phone and relay communication between Layer 3 applications and baseband firmware over serial. we can load the firmware layer1.bin, a simple GSM layer 1 proxy, to the phone, which communicates over the L1A_L23_Interface. It will allow us to run a GSM implementation on our host virtual machine, communicating through the phones radio interface. The last component that needs to be deployed and configured is the mobile L2&3 application.

2.5.3 Link Forwarder Sub Domain

The last phase of our CSS deployment is to enable the link component that will connect the CSS MS sub domain with the CSS network sub domain. The component that can execute the aforementioned task is a PBX in which can be connected the mobile L2&3 application as well as the osmoMSC component through the Osmo-sip-connector which is able to translate between MNCC and SIP protocols. The MNCC protocol has been created by the Osmocom community and allows to control the call handling and audio processing by an external application such a PBX service. With the right configuration on the Astersik PBX service, it will be feasible to forward communication data between osmoMSC component and mobile L2&3 application.

CONCLUSION

In this thesis the security of GSM Um Interface is assessed. Firstly, the GSM security issues, regarding the radio interface, are introduced and an overview of the GSM network architecture is presented. It is proved that GSM technology is still vulnerable to active attacks by performing LTE jamming, call interception and sms impersonation in our LAB environment.

Our lab environment is just a Proof of Concept that pinpoints the security issues of GSM air interface. A more intelligent rogue system (CSS) that we have defined and analysed in the final chapter, can exploit the same GSM security issues that we have already performed, and conduct a Man in the Middle Attack between the victim's mobile device and the service provider's legitimate tower. The MitM Attack exploits the by-design flaw of GSM technology, which allows every mobile device to be downgraded to GSM and camp to a rogue Base Station due to the strongest transmission signal. Furthermore, the rogue CSS, after it dispatches the IMSI from the victim's mobile device, it contacts the legitimate tower with a Location Update request and by using the victim's IMSI. The legitimate network responds with the SRES challenge in order Authentication process to be performed. The rogue CSS forwards the SRES to the victim's mobile device to be calculated. The mobile device solves the challenge, forwards it to the rogue CSS, which then forwards it back to the legitimate network. After it authenticates the CSS, it requests an encryption method and the CSS responds with A5/0. Finally, the network accepts the rogue CSS's connection request. In this thesis the MitM attack is defined and analysed thoroughly as well as the components that assemble the rogue CSS system.

Lastly, as we have already discussed in the beginning of this thesis, every system is as secured as its weakest component. GSM is still the most popular mobile telecommunication technology globally and it will remain for the foreseeable future. Although most of the security issues seem to not be resolved yet. As long as the GSM technology remains vulnerable and exploitable, we cannot be confident for the security of the newer mobile network technologies.

REFERENCE

[1] From GSM to LTE-Advanced, 2nd Edition – Wiley

[2] https://blogs.windows.com/devices/2013/05/22/why-gsm-networks-still-matter/

[3] https://wiki.myriadrf.org/LimeSDR-Mini

[4] https://www.elektormagazine.com/news/review-lime-sdr-mini

[5] https://greatscottgadgets.com/hackrf/one/

[6] https://wiki.myriadrf.org/Lime_Suite

[7] https://www.mcc-mnc.com/

[8] https://github.com/myriadrf/LimeSuite

[9] Getting Started with OpenBTS – O'REILLY

[10] OpenBTS-4.0-Manual – Range Networks

[11] https://osmocom.org/projects/osmotrx/wiki/OsmoTRX

[12] https://ftp.osmocom.org/docs/latest/osmotrx-lms-vty-reference.pdf

[13] https://osmocom.org/projects/osmotrx/wiki/LimeSDR_Family?fbclid=IwAR2-nhUlQm73x9CY9j_BTxPkwDevBmpgAvXUD5N_YalOsCvqzAt_w-v1aUE#Running-osmo-trx-lms-with-LimeSDR

[14] https://discourse.myriadrf.org/t/openbts-1st-test/823/11

[15] https://www.youtube.com/watch?v=CYO-SKfdtHw&list=PLnzEbgyK52Gu9fdVDHburrsG3KBIntXFK&index=47

[16] https://arxiv.org/pdf/1002.3175.pdf

[17] https://hal.inria.fr/hal-01480210/document

[18] https://en.wikipedia.org/wiki/Software-defined_radio

[19] https://www.sciencedirect.com/topics/engineering/software-defined-radio

[20] https://www.voip-info.org/mixmonitor/

[21] https://wiki.asterisk.org/wiki/display/AST/Asterisk+11+Application_MixMonitor

[22] https://nickvsnetworking.com/gsm-with-osmocom-part-3-bts-in-practice-with-limesdr-osmo-bts-trx/

[23] https://www.youtube.com/watch?v=fQSu9cBaojc

[24] https://www.eff.org/wp/gotta-catch-em-all-understanding-how-imsi-catchers-exploit-cell-networks

[25] https://www.slideshare.net/deepakecrbs/overview-of-gsm-cellular-network-amp-operations

[26] https://www.researchgate.net/figure/GSM-protocol-stack-The-protocol-stack-is-composed-of-3-layers-Layers-1-and-2-are-the_fig1_234782039

[27] https://www.etsi.org/deliver/etsi_gts/04/0404/03.03.04_60/gsmts_0404sv030304p.pdf

[28] https://www.etsi.org/deliver/etsi_gts/04/0401/05.00.00_60/gsmts_0401v050000p.pdf

[29] https://www.etsi.org/deliver/etsi_gts/04/0405/03.01.05_60/gsmts_0405sv030105p.pdf

[30] https://www.itu.int/en/ITU-D/Statistics/Documents/facts/FactsFigures2020.pdf

[31] https://www-users.cs.umn.edu/~hoppernj/celluloc.pdf

[32] https://www.researchgate.net/publication/283559421_Attacking_GSM_Networks_as_a_Script_Kiddie_Using_Commodity_Hardware_and_Software

[33] https://en.wikipedia.org/wiki/Software-defined_radio

[34] https://www.sciencedirect.com/topics/engineering/software-defined-radio

[35] https://www.gnuradio.org/about/

[36] https://www.researchgate.net/publication/318579182_Cross_layer_attacks_on_GSM_mobile_networks_using_software_defined_radios

[37] https://www.kathimerini.gr/society/561294997/mallon-ton-skotosame-o-dialogos-poy-kategrapse-i-antitromokratiki-gia-tin-epithesi-ston-astynomiko/

[38] https://theintercept.com/2020/07/31/protests-surveillance-stingrays-dirtboxes-phone-tracking/

[39] https://osmocom.org/

[40] https://osmocom.org/projects/osmobts

[41] https://osmocom.org/projects/osmotrx/wiki/OsmoTRX

[42] https://osmocom.org/projects/osmobsc/wiki

[43] https://osmocom.org/projects/osmomsc/wiki

[44] https://osmocom.org/projects/baseband/wiki

[45] https://osmocom.org/projects/baseband/wiki/Phones

[46] https://osmocom.org/projects/baseband/wiki/HardwareCalypso

[47] https://osmocom.org/projects/baseband/wiki/Serial_Cable

[48] https://osmocom.org/projects/baseband/wiki/Osmocon

[49] https://osmocom.org/projects/baseband/wiki/Software

[50] https://osmocom.org/projects/baseband/wiki/Mobile

[51] https://www.eff.org/wp/gotta-catch-em-all-understanding-how-imsi-catchers-exploit-cell-networks

[52] https://osmocom.org/projects/baseband/wiki/Branches