



## Different Vulnerabilities of GSM Network

Kunal M. Pattani<sup>1</sup>, Arjunsinh Parmar<sup>2</sup>

<sup>1</sup>Asst. Professor, Dept. of E&C, C U Shah College of Engg. & Tech., Wadhwan, Gujarat, India

<sup>2</sup>P.G.Student, Dept. of E&C, C U Shah College of Engg. & Tech., Wadhwan, Gujarat, India.

**Abstract**—Global System for Mobile communications (GSM) is the most popular telecommunication protocol used in telecommunication networks. The telecommunications industry uses a combination of 2G (GSM), 3G (UMTS) and 4G (LTE) systems to access communication worldwide. However telecommunications industry keeps a high percentage of their deployed infrastructure using GSM technologies. GSM offers worldwide roaming and interconnection with any available GSM network. GSM subscriptions have the 82.4% percent of the global market of mobile communications serving more than 4 billion people across the planet. Users are expected to be aware of the possible security threats. This work highlights weaknesses and issues in the GSM standard, and presents an informed approach to help audit GSM networks for vulnerabilities.

**Keywords**- GSM Vulnerability; Sniffing GSM; RTL-SDR; Kali Linux; GSM attack; Security; Privacy.

### I. INTRODUCTION

The concept of GSM emerged from a cell based mobile radio system at Bell Laboratories in the early 1970s. GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard. GSM is the most widely accepted standard in telecommunications and it is implemented globally. As of 2014 it has become the de facto global standard for mobile communications - with over 90% market share, operating in over 219 countries and territories [4]. GSM was developed using digital technology. GSM network developed as a replacement for first generation (1G) analog cellular networks, and the GSM standard originally described a digital, circuit-switched network optimized for full duplex voice telephony. This expanded over time to include data communications, first by circuit-switched transport, then by packet data transport via GPRS (General Packet Radio Services) and EDGE (Enhanced Data rates for GSM Evolution). Subsequently, the 3GPP developed third-generation (3G) UMTS standards followed by fourth-generation (4G) LTE Advanced standards. Despite the rapid change in cellular technologies, Mobile Network Operators (MNOs) keep a high percentage of their deployed infrastructure using GSM technologies. With about 3.5 billion subscribers, GSM remains as the only standard for cellular communications [1]. However, the security criteria envisioned 30 years ago, when the standard was designed, are no longer sufficient to ensure the security and privacy of the users. Furthermore, even with the newest fourth generation (4G) cellular technologies starting to be deployed, these networks could never achieve strong security guarantees because the MNOs keep backwards compatibility given the huge amount of GSM subscribers. Recent research has shown that mobile devices data can be used as an effective way to track individuals [2]. This presents a problem related to users' privacy, as their location allows the carriers to profile and track their movement(s). The aim and objective of this paper is to produce knowledge base about publicly available on market equipment that can offer trusted solutions to mobile security researchers to perform penetration tests on mobile network infrastructures. Furthermore, it comes to underline the simplicity with a malicious attacker by only having on its possession a set of low-cost equipment, can introduce an asymmetric threat to a large scale of users without the latter become aware of that being directed. Finally, our paper points those proven vulnerabilities that exist over the years on the worldwide and old GSM network and the lack of security measures and mechanisms that could protect the GSM subscribers.

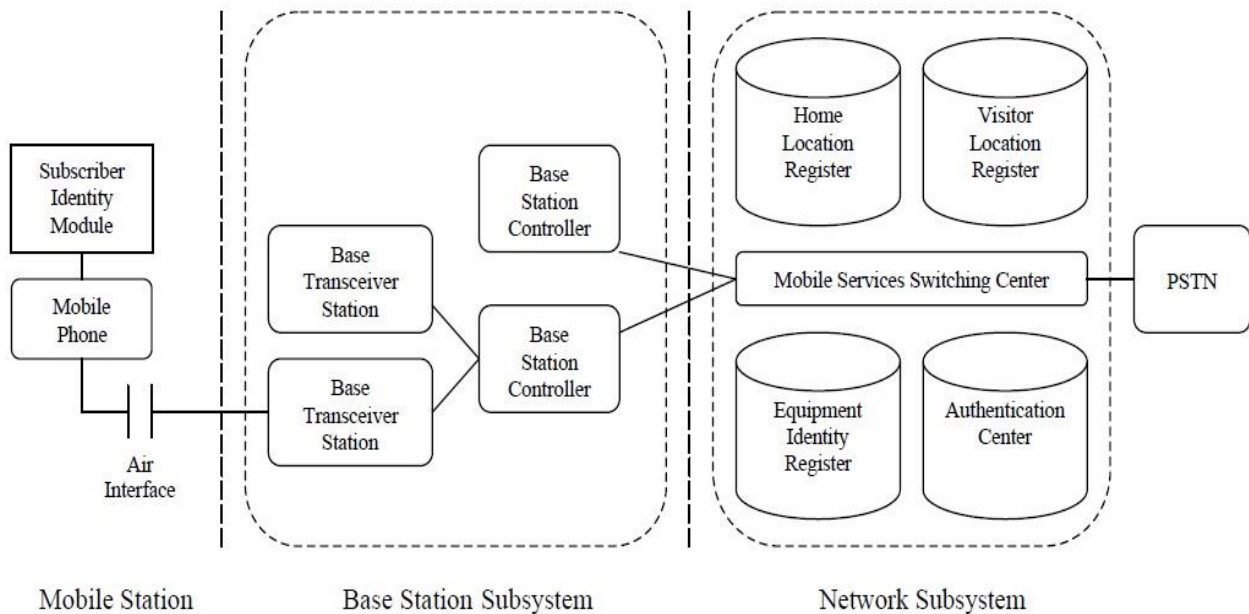
### II. BRIEF BACKGROUND ON GSM

GSM is a very well-known cellular standard, so we only provide a very brief background on some aspects of particular relevance for our work in this section. Fig. 1 illustrates a simplified GSM network architecture. It consists of three major interconnected subsystems that interact between themselves and with the users through certain network interfaces.

The subsystems are:-

- a) Base Station Subsystem (BSS)
- b) Network and Switching Subsystem (NSS)
- c) Operation Support Subsystem (OSS)

The Mobile Station (MS) is also a subsystem, but is usually considered to be part of the BSS for architecture purposes. Equipment and services are designed within GSM to support one or more of these specific subsystems.



**Fig -1:** GSM network architecture

a) Base Station Subsystem (BSS)

The BSS is in charge of providing connectivity between the mobiles and the network. It consists of the Mobile Station (MS), the Base Transceiver Station (BTS), and the Base Station Controller (BSC). The MS is used to provide the user an interface to communicate with the GSM network. It includes the mobile equipment (ME) and the Subscriber Identity Module (SIM). The SIM is used to provide the identity of the user to the network. The BTS transmits and receives the signals from the MSs and controls the transmission power, modulation, voice coding/decoding and encryption of the signals[1]. The BSC controls a set of BTSs as well as the handover, radio channels, paging and other control functions.

b) Network and Switching Subsystem (NSS)

The NSS is in charge of the switching functions, locating the MSs and the interconnection with other networks. It consists of the Mobile Switching Center (MSC), the Home Location Register (HLR), the Visitor Location Register (VLR), and the Gateway Mobile Switching Center (GMSC). The MSC is the main element in the NSS, it controls different BSCs and it is responsible for routing incoming/outgoing calls and for the mobility functions of the terminals such as registration and location of the MSs. The HLR is a static database that contains specific parameters of the subscriber (location information, authorized services, type of terminal, etc).The VLR is a dynamic database and it is associated with one MSC, it stores information of the terminals that are registered with the MSC. When a MS registers with the network, the corresponding VLR verifies the different parameters with the HLR of the home network. The GMSC is the interconnection point between the GSM network and external networks for which it provides gateway functions.

c) Operation Support Subsystem (OSS)

The OSS controls, in a centralized manner, the management and maintenance of the GSM subsystems. It consists of the Authentication Center (AuC), and the Equipment Identity Register (EIR). The AuC contains a database that stores the identification and authentication of every subscriber. It stores the International Mobile Subscriber Identity (IMSI) and the permanent key associated with every SIM (Ki).The EIR is a database that stores lists of the MSs identified by their International Mobile Station Equipment Identity (IMEI). It is used to determine if the MSs are authorized, unauthorized or in need to be monitored.

### III. SECURITY IN GSM

GSM security is addressed in two aspects: authentication and encryption. Authentication avoids fraudulent access by a cloned MS. Encryption avoids unauthorized listening.

A secret key,  $K_i$ , is used to achieve authentication.  $K_i$  is stored in the AuC as well as in the SIM. The  $K_i$  value is unknown to the subscriber. To initiate the authentication process; the home system of the MS generates a 128-bit random number called RAND. This number is sent to the MS. By exercising an algorithm, A3, both the network (AuC) and the MS (SIM) use  $K_i$  and RAND to produce a signed result (SRES). The SRES generated by the MS is sent to the home system and is compared with the SRES generated by the AuC. If they are not identical, the access request is rejected. Note that if the SRES and RAND generated by the AuC are sent from the HLR to the visited VLR in advance, the SRES comparison can be done at the visited VLR. Algorithm A3 is dependent on the GSM service provider. Since the visited system may not know the A3 algorithm of a roaming MS, authentication result SRES is generated at the home system of the MS[10].

If the MS is accepted for access, an encryption key produced by an algorithm, A8, with  $K_i$  and RAND as inputs. Like A3, A8 is specific to the home system has generated  $K_c$ , this encryption key is sent to the visited system.  $K_c$  and the TDMA frame number encoded in the data-bits are used by an algorithm, A5, to cipher and decipher the data stream between the MS and the visited system. The same A5 algorithm may be used in all systems participating in the GSM service[10].

The cellular service provider has track the location of mobile subscribers in a efficient way by making competent use of the radio resources. In order to accomplish that, the large areas that being served from a cellular network are parted into smaller geographical regions like the well-known Location Areas (LA, LAC). Then, the broadcast messages will be addressed in those smaller areas. Identifying the paging requests that carry TMSIs of the users, we can suppose if an individual resides in that area in case we know the specific temporary ID. Moreover, the temporary ID is the only identifier by observing the broadcasted messages of paging procedure so it could be a difficult procedure to map the temporary ID with the telephone number of the user.

From the GSM specifications and from mobile network operators is strict policy is considered that the IMSI must sent as rarely as possible, to avoid it being located and tracked. However by reviewing the above and as it observed during our experiments and attacks, there multiple times that network authenticates its users by the IMSI.

Across the history of the GSM standard, there have been many attacks to the protocol. In 1998, reverse engineering techniques were applied to break the 3GPP subscriber authentication algorithms implementation [3]. Since then, numerous attacks to the different versions of the encryption algorithms have been reported in [5], [6] and [7].

### IV. DIFFERENT VULNERABILITIES OF GSM NETWORK

#### 4.1 Sniffing GSM Traffic

In this section, we describe our scenario, the tools needed to perform the attack and we detail the implementation of the attack.

##### 4.1.1 Tools

We now briefly describe the set of tools used to perform the attack:

**Kali Linux OS (2016.2, 64-bit):** Kali Linux is a Debian-derived Linux distribution designed for digital forensics and penetration testing. It is maintained and funded by Offensive Security Ltd. Mati Aharoni, Devon Kearns and Raphaël Hertzog are the core developers. Kali Linux is preinstalled with over 300 penetrationtesting programs, including Armitage (a graphical cyberattack management tool), nmap (a port scanner), Wireshark (a packet analyzer), John the Ripper (a password cracker), Aircrack-ng (a software suite for penetration-testing wireless LANs), Burp suite and OWASP ZAP (both web application security scanners).Kali Linux can run natively when installed on a computer's hard disk, can be booted from a live CD or live USB, or it can run within a virtual machine. It is a supported platform of the Metasploit Project's Metasploit Framework, a tool for developing and executing security exploits[11].

**Wireshark:** Wireshark is a network analysis tool previously known as Ethereal. It captures packet in real time and display them in human readable format. Basically, it is a network packet analyzer which provides the minute details about your network protocols, decryption, packet information, etc. It is an open source and can be used on Linux, Windows, OS X, Solaris, NetBSD, FreeBSD and many other systems. The information that is retrieved via this tool can be viewed through a GUI or the TTY mode TShark Utility.

**Airprobe:** Airprobe is a GSM air interface analysis tool [9].

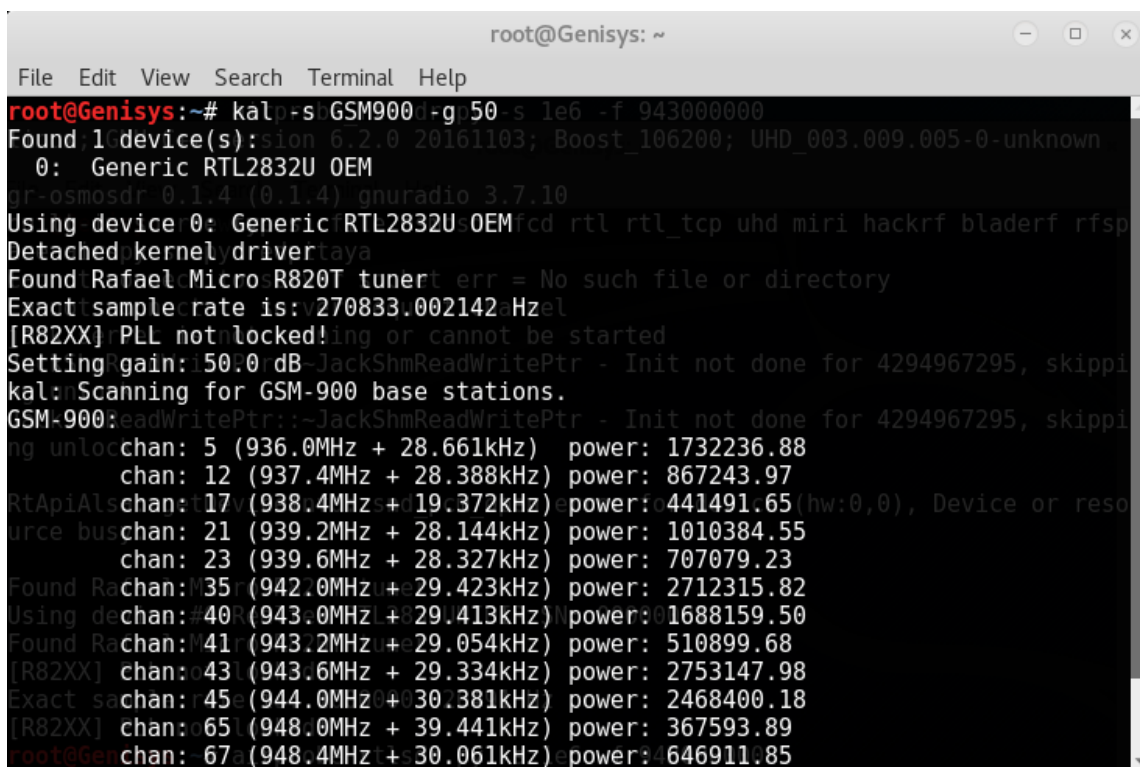
**Kalibrate (kal):** It is an open-source software project used to scan the GSM frequencies of the base stations in the vicinity and capable of determining the local oscillator frequency offset [12].

**GNU Radio:** It is an open-source toolkit that offers real-time signal processing as well as the possibility to implement different radio technologies.

**RTL-SDR Dongle:** RTL-SDR is a special commodity hardware that consisted to be as wideband software defined radio (SDR) scanner. RTL can be used with a DVB-T TV Tuner dongle. RTL-SDR is a very broadband (60MHz to 1700MHz) product and has a large scale of applications on different things. RTL can be used as a telecommunication “antenna” for TV broadcasting.

#### 4.1.2 Implementation

Beginning with the RTL-SDR we have to install the Kalibrate utility. Kalibrate is a useful tool that enables us to identify the available principal GSM channels in our area. Kalibrate-RTL or kal is a Linux program used to scan for GSM BTSs in a given frequency band. The first thing is to find out what frequencies we have GSM signals in our area. For most of the world, the primary GSM band is 900 MHz. Using the kalibrate utility we execute the following as shown in Fig -2:



```
root@Genisys: ~
File Edit View Search Terminal Help
root@Genisys:~# kal -s GSM900 -g 50 -s le6 -f 943000000
Found 1 device(s): sion 6.2.0 20161103; Boost_106200; UHD_003.009.005-0-unknown
0: Generic RTL2832U OEM
gr-osmosdr 0.1.4 (0.1.4) gnuradio 3.7.10
Using device 0: Generic RTL2832U OEM fcd rtl rtl_tcp uhd miri hackrf bladerf rfsp
Detached kernel driver taya
Found Rafael Micro R820T tuner err = No such file or directory
Exact sample rate is: 270833.002142 Hz
[R82XX] PLL not locked! ping or cannot be started
Setting Rgain: 50.0 dB - JackShmReadWritePtr - Init not done for 4294967295, skipping
kal: Scanning for GSM-900 base stations.
GSM-900: readWritePtr::~JackShmReadWritePtr - Init not done for 4294967295, skipping
chan: 5 (936.0MHz + 28.661kHz) power: 1732236.88
chan: 12 (937.4MHz + 28.388kHz) power: 867243.97
chan: 17 (938.4MHz + 19.372kHz) power: 441491.65 (hw:0,0), Device or resource busy
chan: 21 (939.2MHz + 28.144kHz) power: 1010384.55
chan: 23 (939.6MHz + 28.327kHz) power: 707079.23
Found Ra chan: 35 (942.0MHz + 29.423kHz) power: 2712315.82
Using de chan: 40 (943.0MHz + 29.413kHz) power: 1688159.50
Found Ra chan: 41 (943.2MHz + 29.054kHz) power: 510899.68
[R82XX] chan: 43 (943.6MHz + 29.334kHz) power: 2753147.98
Exact sa chan: 45 (944.0MHz + 30.381kHz) power: 2468400.18
[R82XX] chan: 65 (948.0MHz + 39.441kHz) power: 367593.89
root@Gen chan: 67 (948.4MHz + 30.061kHz) power: 646911.85
```

Fig -2: Scanning BTS Frequencies

Now, these are the frequencies that we have to tune with our RTL-SDR dongle to start capturing GSM packets. Fig.-3 shows the operation of RTL-SDR. Now it's time to start capturing the downlink GSM traffic generated from a specific BTS. From that purpose we have to execute “airprobe\_rtlsdr” python script included in the libsmocore / airprobe library. Firstly we find the path the script located after execute it giving as attribute one of previous found available frequencies. In order to have better results regarding the captured traffic it is better to use the frequency with the best power (HZ). Now we execute the python script by command “airprobe\_rtlsdr.py -s le6 -f 953000000”. Here we choose 953 MHz frequency. Then a screen is displayed with the spectrum captured in real-time. Alongside, we start Wireshark on a new terminal window. Airprobe dumps data into a UDP port, so we had to set Wireshark to listen to this. Under the Start button in Wireshark, we first set the capture interface to Loopback: lo and then pressed Start.



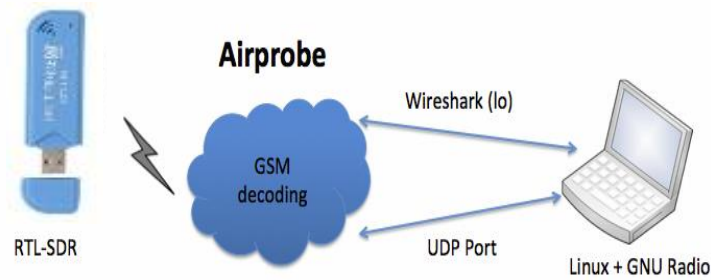


Fig -3: Operation of RTL-SDR

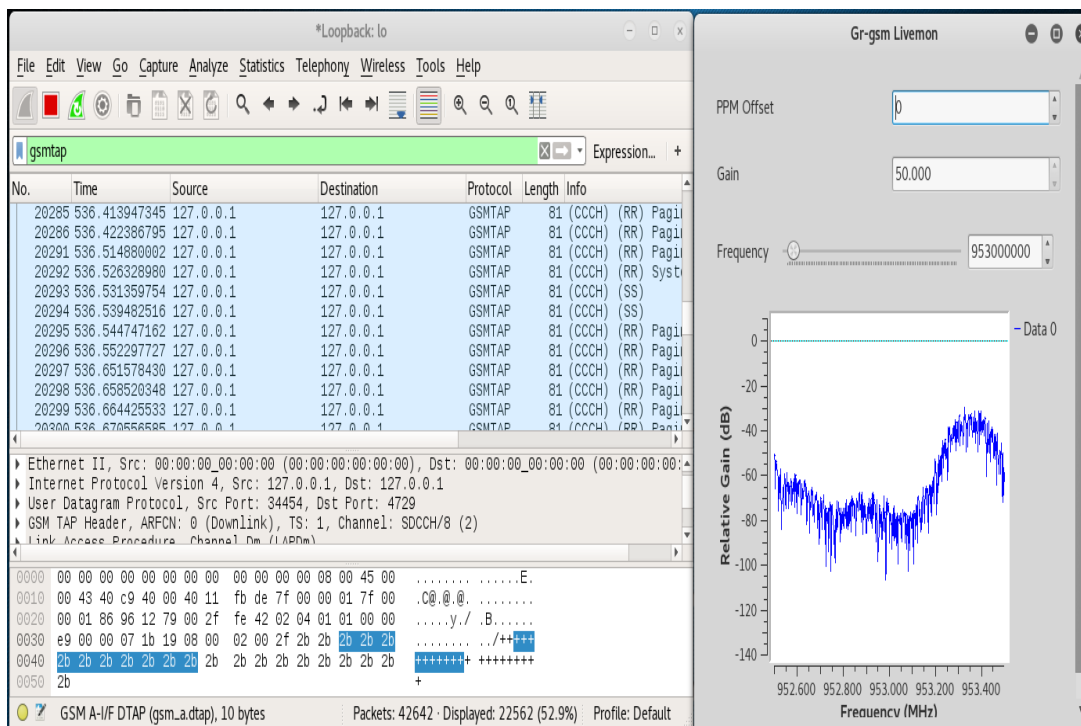


Fig -4: Capturing GSM live traffic

Stopping the `airprobe_rtlsdr.py` we continue with the analysis of the packet capture files. As we observe from the screenshots as shown in Fig -4, packets transmitted from BTS to different MSs named as Paging Request or System information, each one of this contains different information. Here is a brief analysis from each one of the captured messages:

### System Information Message

We start our analysis from System Information messages. Generally this type of message contains the info that MS needs in order to communicate with the network. As we can see there are different types of such messages each one contains various piece of information.

**Type 1:** Channel type = BCCH: Contains a list of ARFCN(Absolute Radio Frequency Channel Number)s of the cell and RACH control parameters.

**Type 2:** Channel type = BCCH: Contains neighbor cell description (list of ARFCNs of the cell) and BCCH frequency list.

**Type 3:** Channel type = BCCH: Contains cell identity (cell ID) code decoded, Location Area Identity-LAI (which involves Mobile Country Code (MCC), Mobile Network Code (MNC) and Location Area Code (LAC)) and some GPRS information.

**Type 4:** Channel type = BCCH: Contains LAI (MCC+MNC+LAC) decoded, Cell selection parameters and RACH control parameters. Some GPRS information too.

**Type 2ter:** Channel type = BCCH: Contains neighbor cell description (list of ARFCNs of the cell) with Extended BCCH frequency list.

**Type 2quarter:** Channel type = BCCH: Is 3G message with information that we don't take into account in this study. Contains 3G-neighbor cell description.

**Type 13:** Channel type = BCCH: They contain all the important information about GPRS like GPRS Cell options and GPRS power control parameters.

**Paging Request Message**

**Type 1:** Channel type = CCCH  
 Contains: Mobile Identity 1 number (IMSI)  
 Page Mode = normal paging (P1)  
 Channel Needed.  
 Contains: Mobile Identity 1 and 2 = TMSI/P-TMSI  
 Page Mode = normal paging (0)  
 Channel Needed

**Type 2:** Channel type = CCCH  
 Contains: Mobile Identity 1, 2 = TMSI/P-TMSI or IMSI  
 Mobile Identity 3  
 Page Mode = normal paging (0)  
 Channel Needed

**Type 3:** Channel type = CCCH  
 Contains: Mobile Identity 1, 2, 3 and 4 = TMSI/P-TMSI (Not decoded)  
 Page Mode = normal paging (0)  
 Channel Needed

**Immediate Assignment Message**

Channel type = CCCH  
 Contains: Time Advance Value  
 Packet Channel Description (Time Slot)  
 Page Mode = Extended Paging (1)

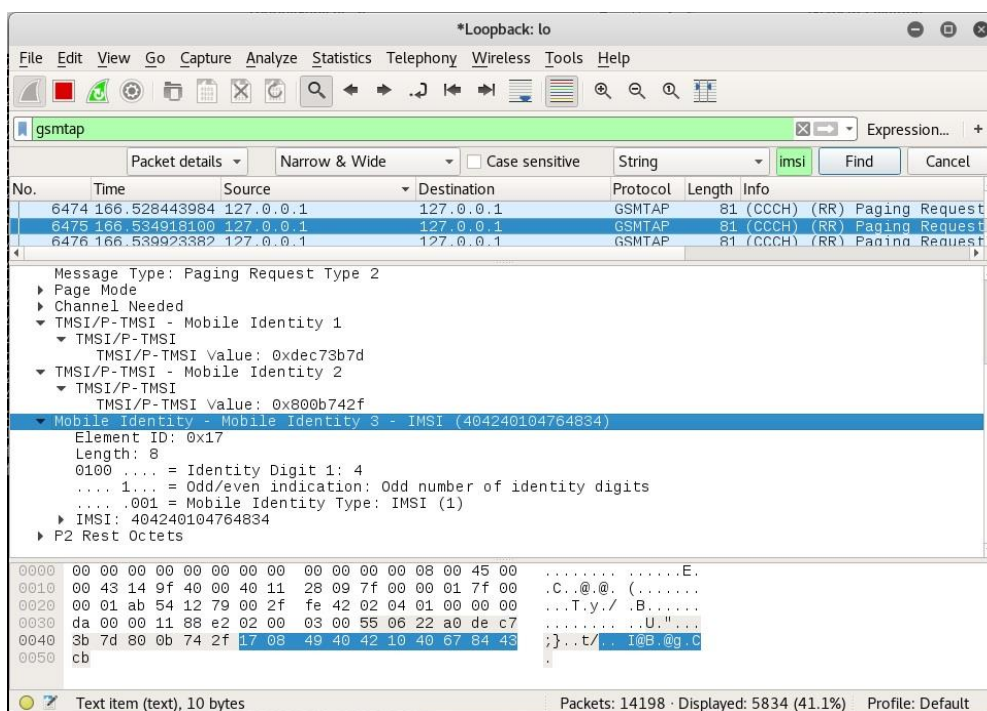


Fig -5: Paging Request Message-IMSI

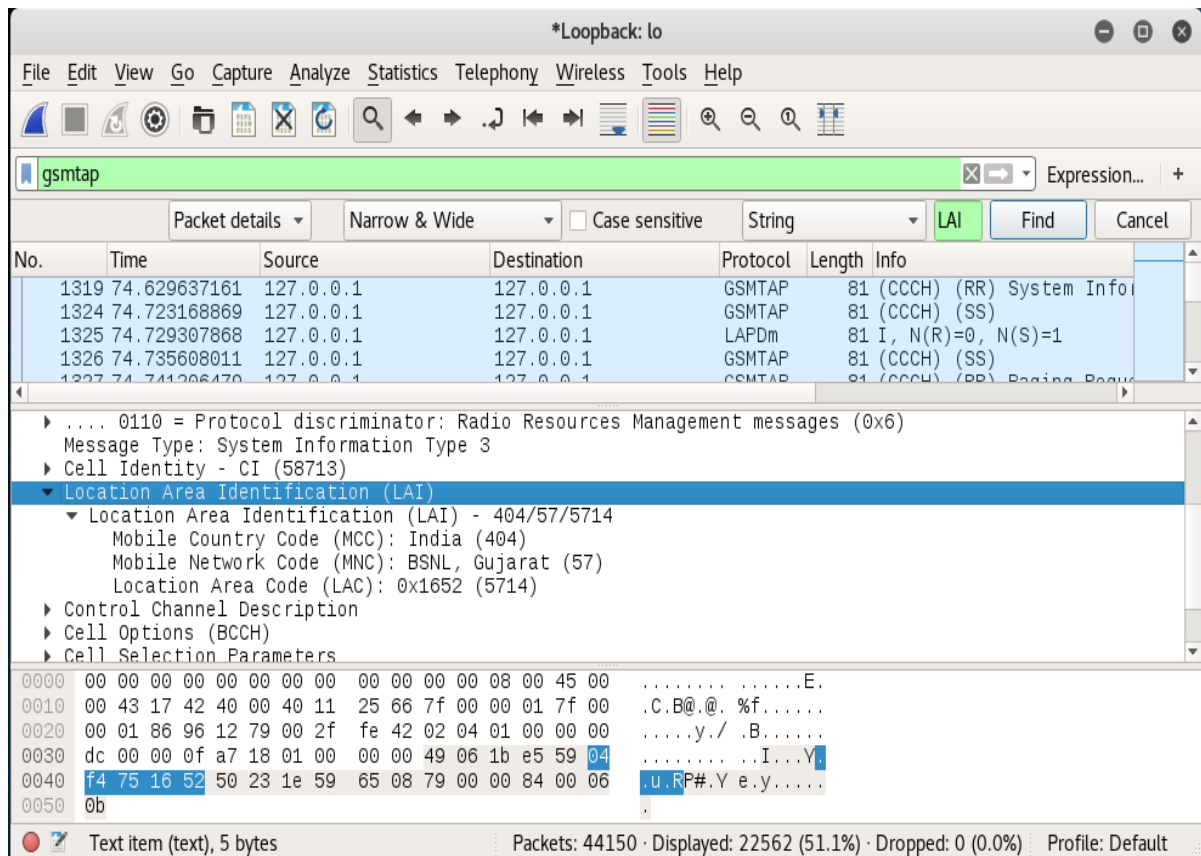


Fig -6: Cell ID and LAI Information from Wireshark Captured packets

As we can show in Fig-5 and Fig-6, it gives the IMSI, TMSI, Ciphering algorithm (A5/1 or A5/2 or A5/3), Cell Id, LAI, MCC, MNC, LAC, etc.

IMSI actually represents the unique identity for the subscriber of the phone including the origin country and mobile network that the subscriber subscribes. It basically identifies the user of a cellular network and every cellular network has its own unique identification. Basically, all GSM networks use IMSI as the primary identity of a subscriber or user. The number that represents IMSI can be as long as 15 digits or shorter. The first three digits are the Mobile Country Code (MCC) and followed by the Mobile Network Code (MNC). The information of IMSI is also contained in the SIM card. IMSI are normally used by network operator to examine the subscribers and whether to allow the subscriber to use another network operator. By tracking your IMSI, the authority can actually track not just the location of your phone but also who you are calling, at what time and where the call is made.

Each location area of a Public Land Mobile Network (PLMN) has its own unique identifier which is known as its Location Area Identity (LAI). This internationally unique identifier is used for location updating of mobile subscribers. It is composed of a three decimal digit Mobile Country Code (MCC), a two to three digit Mobile Network Code (MNC) that identifies a Subscriber Module Public Land Mobile Network (SM PLMN) in that country, and a Location Area Code (LAC) which is a 16 bit number thereby allowing 65536 location areas within one GSM PLMN.

The LAI is broadcast regularly through a Broadcast Control CHannel (BCCH). A mobile station (e.g. cell phone) recognizes the LAI and stores it in the Subscriber Identity Module (SIM). If the mobile station is moving and notices a change of LAI, it will issue a location update request, thereby informing the mobile provider of its new LAI. This allows the provider to locate the mobile station in case of an incoming call. So we can say that this information is very sensitive to the privacy and security of mobile phone users.

## 4.2 Stealthy Denial of Service (DoS) Attack To Mobile Station:

Here we perform a denial of service attack to a specific MS by using the Arduino microcontroller combined with the GSM shield. The effectiveness of such an attack is that the victim MS will be unable to receive and serve legitimate phone calls. The way in which this attack can be accomplished is that we continuously make original PSTN calls towards a target MS by using Arduino scripting capabilities where the latter can simulate phone calls in a repetitive mode. The meaning and the key of this attack is that the victim cannot identify that it is under attack due to the fact that the mobile phone does not actually ring. For that reason this type of denial of service can be characterized absolutely as a “stealthy” attack.

### 4.2.1 Examining An Original Phone Call :

Before explain and analyze the basic steps that followed in order to accomplish the DoS attack, it has separate meaning to understand the sentence: “the target phone does not actually ring”. The time we started making various phone calls using the custom scripts written in Arduino IDE, observed that there a quite small time space by the time we dial a MS and the time the MS rings. Launching a plenty of original calls and after applying different counters and AT commands in order to repeat and after terminate a phone a call we reached to an important conclusion. Finally we wrote a custom script that confirms the following conclusion.

As you can see from the following figure the time sequence of an original phone call in a cellular network, identified that consists of three separate phases that occur successively.

1. Phone dialing
2. Paging request
3. Phone ringing

More specifically, at the first time period ( $t_0$ ) we dial the phone number of the MS we want to attack. Next, at  $t_1$  the network has just started the paging procedure in order to locate the victim MS and transmit our call request. It is exactly the time space in which the BTS transmit the paging request message to the target MS. Keep notice in that whole time space ( $t_0 - t_1$ ) the MS does not ring. The important issue here is that from  $t_1$  and later and after the successful termination of the paging procedure, the MS is considered to be “busy” and any paging request initiated from other callers will be rejected. By reaching the  $t_2$  (after 2.5 seconds) the MS starts to ring. To sum up all the above, the elapsed time between dialing the phone number and the completion of the paging procedure is 3 seconds approximately ( $t_1 - t_0 = 3$  sec). Furthermore, between the paging procedure and phone ringing is on average 2.5 seconds ( $t_2 - t_1 = 2.5$  sec). Generally total time from the moment that a phone number is dialed until the phone rings is on average 5.5 seconds ( $t_2 - t_0 = 5.5$  sec). All these observations regarding the time differences between the three phases during a phone call, were all experimentally tested several times. Anyone can investigate and reach to this conclusion by performing multiple phone calls by using Arduino microcontroller. Whole process is shown in Fig-7.

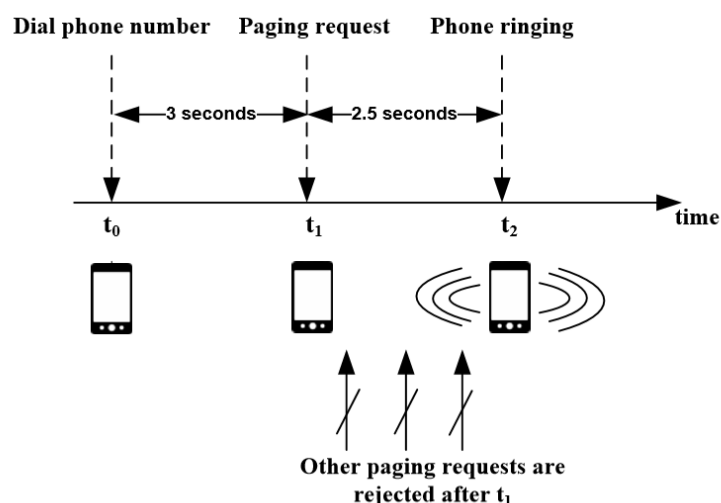


Fig -7: Call Setup - Time Sequence



#### 4.2.2 Denial of Service Attack Analysis :

Taking in consideration the above, we proceed to a stealthy DoS attack targeting a MS. The first action was to insert a SIM card and connect the GSM shield with Arduino. Then plugged it to a laptop with Arduino IDE pre installed via USB connection. Next step was to start developing a custom method taking advantage of the AT commands and specialized functions that included in Arduino IDE GSM libraries. We developed a custom script that had the ability to make recurrently phone calls to a MS including specific counter  $c$ , which was the limit in seconds after which the calls terminated. We set the value of counter accordingly ( $c < 5.5$  seconds) to avoid a probable phone ringing. Compiling and executing the code, we achieved 100 sequential PSTN calls without make the target MS to ring. An important observation there was that during the program execution, the target MS considered to be busy by rejecting any other legitimate phone call towards it. Furthermore, during our attack, no other call activity observed and in combination with the absence of any tone ringing, the MS owner will not become aware that its equipment is under attack in a real environment cases. Concluding, the repetitive calls were not displayed in MS screen as missed calls and generally, no evidence could be identified from victim's point of view.

### V. CONCLUSIONS

In this paper we presented an effective attack that can exploit chronic and fundamental vulnerabilities that exist in the GSM cellular technology. This attack could also have a serious impact at the latest in use cellular technologies like UMTS and LTE. We learned about new come commodity hardware RTL-SDR. RTL-SDR can also be characterized as an IMSI catcher and when combined with some hardware and software can build a mechanism of mobile user tracking. It is obvious that an individual equipped with that cheap commodity hardware could compromise the GSM subscribers' privacy and perform some serious attacks. So, systems with broadcast paging protocols can leak location information and the leaks can be observed with the available and low cost commodity hardware presented in this paper. All these come to exploit the proven vulnerabilities that exist in GSM network and related with the expose of the user's personal identities over the radio link. This research has shown that with certain tools, a system can be created to audit GSM. It is proved that the current protocols used in radio and wireless systems may not be as robust and secure as originally thought.

Denial of Service attack seems quite simple but simultaneously effective. That can prove that no operational security mechanisms have been applied in mobile network security for mitigating and preventing DoS attacks. Stateful firewalls and intrusion detection systems could examine traffic patterns generated on the radio link and could shape and separate legitimate traffic from malicious one.

### REFERENCES

- [1] Santiago Aragon, Federico Kuhlmann and Tania Villa, "SDR-based network impersonation attack in GSMcompatible networks", 81st Vehicular Technology Conference (VTC Spring) 2015, IEEE, pp. 1-5, ISSN: 1550-2252, May 2015.
- [2] Y-A. D. Montjoye, C. A. Hidalgo, M. Verleysen and V. D. Blondel, "Unique in the Crowd: The privacy bounds of human mobility," in *Sci. Rep.* 3, Article number-1376, Mar. 2013.
- [3] G.I.W.D. Briceno, M. (1998) Implementation of comp128. [Online]. Available: <http://www.scard.org/gsm/a3a8.txt>.
- [4] Kameswara Rao Poranki , Yusuf Perwej and Asif Perwej "The Level of Customer Satisfaction related to GSM in India", *RJSITM*, vol. 04, Number 03, pp. 29-36, ISSN: 2251-1563, January-2015.
- [5] E. Barkan, E. Biham, and N. Keller, "Instant ciphertextonly cryptanalysis of gsm encrypted communication," *Journal of Cryptology*, vol. 21, no. 3, pp. 392-429, 2008. [Online]. Available: <http://dx.doi.org/10.1007/s00145007-9001-y>.
- [6] C. Timberg and A. Soltani. (2013, dec) By cracking cellphone code, NSA has ability to decode private conversations. *The Washington Post*. [Online]. Available: <http://wapo.st/18Hc5fC>.
- [7] C. Paget and N. K., "Gsm: Srsly?" in 26th Chaos Communication Congress: Here be dragons.
- [8] A. Fanan , N. Riley, M. Mehdawi, M. Ammar & M. Zolfaghari, "Comparison of Spectrum Occupancy Measurements using Software Defined Radio RTL-SDR with a Conventional Spectrum Analyzer approach", 23rd Telecommunications Forum Telfor (TELFOR) 2015, IEEE, pp. 200-203, Nov. 2015.
- [9] Airprobe. [Online]. Available: <https://svn.berlin.ccc.de/projects/airprobe/>
- [10] Wireless and Mobile network Architectures, Yi-Bang Lin and Imrich Chlamtac, Wiley-India Edition. October 2000.
- [11] [www.kali.org](http://www.kali.org)
- [12] J. Lackey and S. Makgraf. (2012) Kalibrate README. [Online]. Available: <https://github.com/stevem/kalibrate-rtl>.