Università degli studi di Padova

Dipartimento di Ingegneria dell'Informazione
Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea

A Denial-of-Service Attack to GSM/UMTS Networks via Attach Procedure

Relatore:
prof. Mauro Migliardi

Laureando:
Nicola Gobbo

Co-relatore:
prof. Carlo Ferrari

8 luglio 2013 – A.A. 2012/2013

A Paola,
energia e sprone di ogni progetto.

A Mamma e Papà,
per la stima e fiducia concessami.

Abstract

Mobile Network Operators (MNOs) keep a strict control over users accessing the networks by means of the Subscriber Identity Module (SIM). This module grants the user to access the network, by performing the registration and authentication of the user's device. Without a valid SIM module and a successful authentication, mobile devices are not granted access and, hence, they are not allowed to inject any traffic in the mobile infrastructure.

Nevertheless, in this thesis we describe an attack to the security of a mobile network allowing an unauthenticated malicious mobile device to inject traffic in the mobile operator's infrastructure. We show that using a few hundreds of malicious devices without any SIM module it is possible to inject high levels of signalling traffic in the mobile infrastructure, causing significant service degradation up to a full-fledged Denial-of-Service (DoS) attack.

Contents

List of Figures

Introduction

Mobile phones are one of the most pervasively deployed technology in the world and cellular networks have reached worldwide coverage. On one hand, the evolution from early analog networks to recent 4G LTE solutions has allowed operators to offer new services to their customers. On the other hand, the same evolution has pushed new needs into the customers; such needs have evolved from simple phone calls and Short Message Service (SMS) to internet connections and high speed access to streaming data.

The availability of smartphones with wide touch-screen displays as well as the always-on, high bandwidth IP connectivity have generated a growing set of services and applications ranging from e-mail to remote banking, from e-shopping to music streaming, from video on demand to social geo-localized networks. In turn, the ease of use and the availability of a rich a set of functionalities have instilled into users a growing familiarity and a sense of dependency. This dependency does not exist only for leisurable activities, but has a definite onset also in business and critical tasks. In particular, the last years have seen a significant penetration in govern agencies and public bodies. To this aim, we can cite the recent security certification of Android smartphones by the US Department of Defense [28] that allows the deployment of Dell hardware with Froyo (Android OS v2.2) in the Pentagon. A second example is the adoption of tablet PCs (Apple iPad) by the Chicago hospital and the Loyola University Medical Center in Maywood. Finally, several research projects are focusing on the deployment of health-care services onto the tablet PC platform with widely goals from simple access to medical records [10], to reminders for medication intake [29], to decision support systems [19], to automatic recognition of pathological states [24], to systems for memory support [20]. For these reasons, mobile networks security analysis should emphasize availability along with confidentiality and integrity.

However, the introduction of new technologies cannot be decoupled from the support to legacy ones, since i) a high number of older terminals are still active, and ii) some manufactures keep producing 2G-only phones to satisfy low-end market. For these reasons, each new radio access technology has to be deployed alongside existing ones, leading to hybrid architectures where some network components are shared among different technological infrastructures. This condition is driving operators toward single Radio Access Network (RAN) solutions, causing a cellular site to broadcast signals related to up to 3 different technologies in 5 different frequency bands. Such a composite network architecture co-exists with a design traditionally focused on making mobile networks smarter and smarter, while keeping devices crowding their cells as "dumb" as possible [13, 27]. Today's smartphones are far more intelligent and powerful than their predecessors. However, networks still don't profit from their enhanced processing power; on the contrary they assume the lowest possible capability in order to maintain compatibility with older devices. This assumption results in higher signaling traffic levels between network nodes[1], more complex system management and the early consumption of computational resources, even before ensuring whether requesting device is legitimate or not. This difference in workload between server and requester is a vulnerability that, sometimes, may be exploited to mount a particular type of attack called Denial-of-Service (DoS). Despite this terms refers to very different scenarios, having as a common factor the attempt to make a service unavailable to intended users, a typical DoS consider an attacker flooding a target device, i.e. the server, with cheap and seemingly-legitimate requests. The affected equipment has no means to identify and discard malicious requests so it starts to clog up trying to keep up with the increased load, thus not being able to serve all genuine requests, which results in a perceived service outage by the user.

The complexity of the network structure may hide both unknown and known vulnerabilities. For an interesting survey on threats undermining the world of mobile telecommunication, the reader can refer to [8]. For the case of known vulnerabilities, the true impact on the mobile phone network may have not been sufficiently assessed in a way that is similar to what happens in mobile OSes [7]. To this aim, in this thesis we extend the work by Khan et al. [18] focusing on the *attach phase* of GSM/UMTS protocol and we show that it is possible to mount a complete attack even without hijacking or controlling a large number of user IDs recognized by the network. To achieve our goal, we study the amount of signalling traffic that a dedicated SIM-less device can inject into an operator's core network, by pushing air interface to its design limit. Such activity may obviously disable the signalling capabilities of the cells under attack, causing a local DoS similar to the one that can be achieved with a radio jammer; however, to reach a very critical level of disruption, the generated traffic may be targeted at the HLR, i.e. the database containing information on mobile subscribers. Since this database is a critical component of the core network, an outage of its functionality may cause an interruption of other mobile services too, finally resulting in a whole mobile network DoS. In our study, we leverage the HLR performance measurements conducted by Traynor et al. [26], showing that it is possible to achieve a sufficient service degradation using just GSM technology but, taking advantage also of UMTS and combining network elements load conveniently, it is possible to reduce drastically the number of needed devices, still maintaining the SIM-less feature. This results, although not tested in the real networks, are derived from measure-

[1] `http://connectedplanetonline.com/mss/4g-world/the-lte-signaling-challenge-0919/` (accessed in May 2013).

ments and simulations taken from the available literature as well as theoretical estimations based on protocol descriptions and network behaviour; moreover they represent an actual double improvement if compared to the state of the art: in fact, before our study, attacks with the same disruptive potential were described as requiring both i) more device involved and ii) having access to valid SIM cards.

The remainder of this thesis is structured as follow:

- in Chapter 1 we provide a description of the architecture of GSM/UMTS networks;
- in Chapter 2 we analyse the state of the art in the field and we discuss the results obtained in previous related works;
- in Chapter 3 we describe how it is possible to launch an HLR DoS attack with a number of SIMless devices;
- finally, in Chapter 4, we provide some concluding remarks and we describe the future direction of our study.

Chapter 1

GSM/UMTS network description

Global System for Mobile Communications (GSM) standard (2G) was initially designed to carry efficiently circuit switched voice communications in full duplex, with a main advantage over previous analog generation: all the processing happens in the digital domain. The standard protocol set expanded over time with addictions that, from Mobile Network Operators (MNOs) point of view, require just a software upgrade on already deployed hardware; consumers, instead, need modern and more powerful devices to experiment newly offered services. The first addiction to GSM has been General Packet Radio Service (GPRS) that introduced data delivery alongside of voice communications, in both circuit switched and —the more efficient— packet switched mode. Apart from calls GPRS permits data connection throughputs roughly ranging in the 9–170$k$bps interval; augmenting this modest numbers has been the main target of the second GSM enhancement: Enhanced Data Rates for GSM Evolution (EDGE). EDGE is a backward-compatible extension to GSM/GPRS network that introduce new coding and transmission techniques thus allowing for data rates up to 470$k$bps.

Universal Mobile Telecommunications System (UMTS) is a major update to GSM standard which worth it the third generation (3G) epithet. Instead of other GSM updates like GPRS and EDGE, UMTS requires new base station equipments and new frequency band for its deployment. In respect to 2G technologies it is characterized by greater spectral efficiency and higher throughput bandwidth ranging from 348$k$bps of first UMTS release, called R99, to actual 42Mbps of HSPA+. Bandwidth increment is also what drives marketing during early stages of this new technology; great emphasis has been posed by MNOs on services like mobile TV and video calling but their effort has not really been appreciated by end user: in fact, nowadays the main utilization of 3G networks is for plain internet access. UMTS introduction highly affects the radio access portion of the network, the core part, on the other hand, remained the same as in GSM/GPRS in order to facilitate the switch from old technologies to the new one.

A typical GSM/UMTS Public Land Mobile Network (PLMN) consist at least of the infrastructures depicted in figure 1.1. It is mainly split up in three different portions:

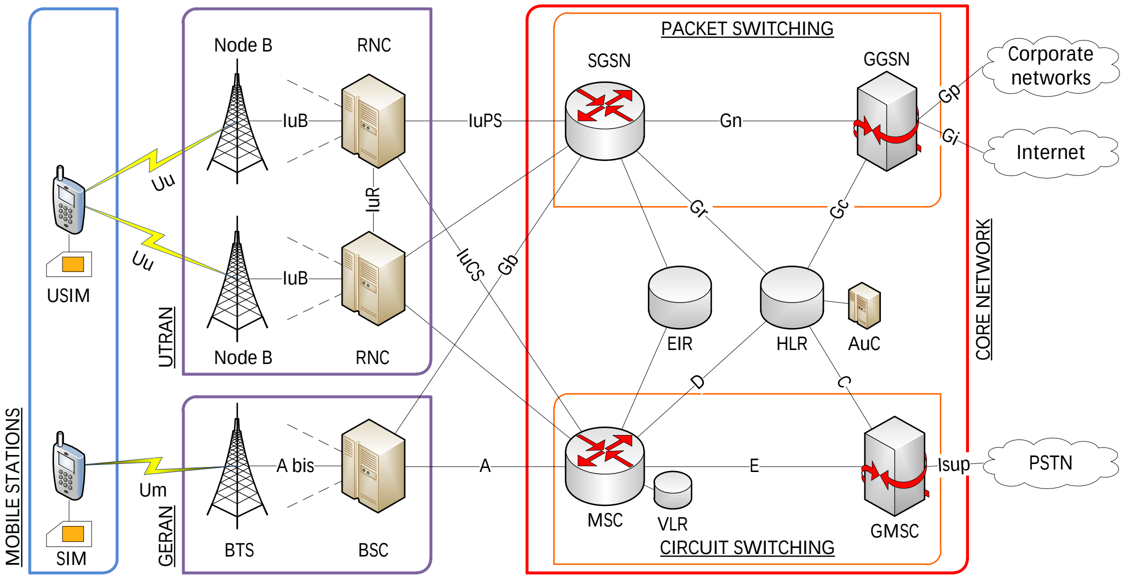


Figure 1.1: GSM and UMTS standard network representation.

- the Mobile Station (MS) or User Equipment (UE);
- the Radio Access Network (RAN) which is called GSM/EDGE Radio Access Network (GERAN) or UMTS Terrestrial Radio Access Network (UTRAN) based on the used technology;
- the Core Network (CN) or Network Switching Subsystem (NSS) with fully separated packet and circuit switched domains.

1.1 The Mobile Station part

MS may be a mobile phone or a mobile broadband modem with appropriate protocol stack and capabilities as defined by specifications. Every device is also marked with a worldwide unique identifier, called International Mobile Equipment Identity (IMEI), that MNOs check against the Equipment Identity Register (EIR), i.e. the database of stolen or out-of-requisites hardware, and, in case of a positive match, banish the faulty equipment from the network.

Nonetheless whichever device is used to connect to the network, there will be a Subscriber Identity Module (SIM) in it. SIMs —or Universal SIM (USIM) in UMTS— are smart cards usually referred to as the furthest extension of mobile operator's network; it securely stores user identity, represented by the International Mobile Subscriber Identity (IMSI), and its related secret key, as long as the algorithms needed during the Authentication and Key Agreement (AKA) phase.

The IMSI is a delicate information because, being unique, allows an eavesdropper to track an user during its movements leveraging unencrypted signalling messages like paging. For this reason, during the preliminary messages exchange after switch on, the user is marked with another identifier, called Temporary Mobile Subscriber Identity (TMSI), that has just a local validity, is often refreshed with a new one, and is used for every communication from and toward the network thus reaching an high degree of anonymity.

1.2 The Radio Access Network part

MSs communicate over air interface with a cell tower that, based on the technology, is called either BTS or Node B. This is the first element composing the RAN, in GSM it has minimum functionality apart from physical layer transmission but, with Node Bs, the trend is toward adding more and more logic to lower response times. A typical BTS/Node B serves three 120°sectors —also called cells— by means of one or more antennas per sector; antennas are powered by amplifiers that gets their pilot signals from one or more baseband modules which are finally connected to the transceiver. Cell towers are grouped together in tens or hundreds and are connected with either a Base Station Controller (BSC) or a Radio Network Controller (RNC). These two devices are the main responsible for the following functions:

radio resource management: this means channel assignments and release as well as MS paging;

mobility management: that, at this level, means inter-BTS/Node B handover;

encryption of user data: these two equipments are the exact point where user informations are encrypted before being sent over the radio interface.

The main difference between BSC and RNC, apart from the protocol they serve, consists in the presence of the `IuR` interface that allows RNC-to-RNC communications: this UMTS novelty, along with the air protocol peculiarities, permits the soft handover, that is, a feature where a cell phone can be simultaneously connected to two or more cells, in order to maximize received signal quality.

1.3 The Core Network part

Each BSC and RNC has a couple of connections toward the core network: one linking the Serving GPRS Support Node (SGSN) carrying packet switched data, the other linking the Mobile Switching Center (MSC) and transporting circuit switched informations. This division come from the fact that GPRS, with its data delivery capabilities, has been a posthumous addendum to the NSS. Both SGSN and MSC act as switching and end point for end-to-end connections it their own domains; they manage hand-overs between different BSC/RNC as well as authentication checking and charging functions. The most valuable operation of these equipments, however, is mobility management: they keep track of MS movements inside their service area and locate it whenever required. To carry out this operation an auxiliary database called Visitor Location Register (VLR) is used: it contains the user identity along with an indication of its current location at the BSC/RNC-level, and a pointer to the MNO's main user record which is contained in another database called Home Location Register (HLR).

The HLR maintains a record for each mobile phone subscriber with details like the telephone number, IMSI and secret key —the same contained in the SIM—, call blocking and forwarding rules and a pointer to the most updated VLR the user is known to be roaming on. HLR is a core component for the networks because it has to be queried for phone call and SMS delivery, billing procedures and authentication: in this last function it is supported by the Authentication Center (AuC) which calculates challenges and responses that will then be sent to the MSC/SGSN for actual user validation.

Chapter 2

Survey of mobile network attacks

Cellular networks seem unaffected by the same threats that, almost daily, come up in the newspapers regarding other types of widely spread systems like the Internet. Nonetheless, even if a large security outbreak has not already made its way through the news, mobile operators' network security has been studied in the literature for quite a long time. Initially, most of the attention of researchers was focused on confidentiality and integrity of data traveling over the wireless portion of the system; however, in more recent works, the problem of the actual availability of the services provided by the network, both in the wireless segment and in the core network segment, has gained popularity, becoming the focus of different studies.

2.1 Jamming attacks

The simplest way to prevent a mobile network from offering its services is using a radio jammer. Xu et al. [30] define four jamming models differing in type and duration of the emitted signal and study the feasibility of detecting such attacks. They show that a jammer always injecting regular data, called deceptive, is the most effective one but the random version, which alternates between sleeping and transmitting, may represent a valid alternative taking energy conservation in consideration. However, even with smart, protocol-specific, jamming algorithms like [23], the intrinsic trade-off between finite power supply and continuous transmission make this kind of attack limited both in space and time.

From a detection point of view Xu et al. [30] conclude that a single performance indicator like, for example, signal strength or packet delivery ratio, is not enough to spot an ongoing jamming attack: thus they define two algorithms based on *classification* and *consistency check* phases that mix together multiple indicators in order to conclude the presence of a jammer.

2.2 Smartphone: the mobile network outlier

Moving from physical towards upper layers increases both the complexity of the attack and the size of the involved network segment. In order to be able to prove higher layer attacks possible, however, researchers have had to wait for a device with extensible capabilities, a kind of device that made its first market appearance in 2000 but actually had a significant deployment only in 2007: the *smartphone*[1]. Until late 1990s mobile phones had only basic phone features so the user had complete control over what the terminals were doing. This fact, however, has been subverted by the first iPhone release in 2007 and, more specifically, by the introduction of Apple App Store. The iPhone, in fact, as all the smartphones marketed today, ran an operating system over which a series of applications offered an open ended set of end-user functionalities (e.g., personal information management, e-mail access, web browsing and much more). Thus users, in a way that is very similar to using traditional PCs, may extend the default application set through vendors' specific application stores where new service-enabling third-party applications can be bought, downloaded and installed. The advent of application-enabled phones and centralized software distribution systems attracted the attention both of attackers[2] and of security researchers. In particular, the research community has proved that the open feature set nature of the smartphone makes it the device capable of massive and distribute mobile network attacks [9].

2.2.1 The disruptive potential of smartphone botnets

Past Internet security studies prove that in order to mount a DoS attack a botnet is the tool that provides the most suitable characteristics; however, mobile networks have constraints and peculiarities that should be taken into consideration. In particular, Fleizach et al. [12] study how "fast" malware may propagate using two fake vulnerabilities, affecting VOIP and MMS reception. They model both a single mobile operator's network topology and different contact graph distributions showing that, by leveraging the generally distributed architecture of VOIP services, a VOIP infection can reach 70% of users in around 4 hours generating major congestion effects on the RNC-to-SGSN link (see previous section). On the other hand, MMS infection spreads at a much slower pace because it is constrained by a few centralized servers that act as bottlenecks.

Creating a mobile phone botnet is generally more challenging than doing it with traditional Internet nodes; this derives both from the fact that mobile phones nodes are usually less apt at running daemon processes and to the fact that most of the time mobile phones are connected to the internet with a private IP address. Furthermore, as Mulliner and Seifert [21] analyze in their study the command and control (C&C) part has non-negligible set of specific challenges. As they point out,

[1] `en.wikipedia.org/wiki/Smartphone` (accessed in May 2013).

[2] `http://arstechnica.com/security/2013/04/family-of-badnews-malware-in-google-play-downloaded-up-to-9-million-times/` (accessed on May 2013).

mobile phones environment forces botnet master to face challenges like limited run time, communication costs and absence of public IP address: all of these specific problems have to be addressed in order to keep the malware concealed to the user. To identify methods to overcome the above mentioned problems, the authors identify three communication approaches: the first based on SMS-only messaging, the second based on IP packet delivery in a peer-to-peer topology and the third based on a SMS-HTTP hybrid design; their analysis allows to conclude that the last one is the most promising and dangerous botnet C&C structure, and it also outlines some communication strategies that would help in keeping low bills.

An attacker capable of controlling a botnet can use infected devices for multiple purposes. Spam delivery is a first possible use. Sending junk or marketing messages through SMS is one of the easiest thing, and the attacker can even get a direct revenue stream by forcing clients to make calls or send SMS to premium price services [11]. Another type of attack stems from the fact that MNOs and users identify a telephone number —that is a SIM card— with a real person identity. Exploiting such a trust link, coupled with the possibility of registering whatever input or conversation make remote wiretapping and identity theft or spoofing [13, 11] straightforward for an attacker. A malicious entity may also try to kick mobile network elements out of service. As an example, Guo et al. [13] predicted that a few dozens of subverted smartphones, served by the same base station, can jeopardize its availability by making no-answer calls and thus saturating provisioned voice channels. If phones are not located in the same place, authors outlined that it is still possible to put call aggregation points to a halt by means of a *distributed* denial of service: the number of needed controlled devices is indeed higher than the one needed in the previous case, but, due to the fact that PSTN, cellular switches and call centres are designed for a limited Busy Hour Call Attempts, the attack is still feasible.

Later studies still focusing on DoS attacks show that it is possible to achieve the needed level of service degradation in a more efficient way: instead of consuming traffic (or user-plane) channels, an attacker may try to flood control channels which are usually separated from traffic ones and significantly more limited in terms of available bandwidth. One of the first work in this direction is from Traynor et al. [25]. In a strict sense, the attack described here doesn't use a botnet but, in a broader sense, every mobile phone is an accomplice because what it has to do is just receiving incoming requests. They show how the interconnection between the mobile network and the Internet via, for example, on-line SMS delivery capabilities, may be exploited by an attacker continuously sending text messages to an especially crafted hit-list of telephone numbers. Such a data flood will keep the GGSM Standalone Dedicated Control Channel (SDCCH) —responsible for authentication and setup of both voice calls and text messaging— saturated with text messages, thus unavailable to accept or delivery any voice call, even with available traffic channels: to prove effectiveness of this type of attacks authors simulate that approximately 580kbps of injected SMS traffic is enough to deny service in the whole Manhattan area.

Another study from Traynor et al. [27] focuses on the GPRS network and characterizes two different types of radio resource exhaustion attacks targeting data connection setup and tear-down mechanisms. In the setup attack authors continue exploring control channel depletion effects but, this time, they analyze the Random Access Channel (RACH). RACH is shared by all mobile terminals attempting to establish connections with the network and, in order to minimize contention, its access is mediated through slotted-ALOHA protocol. During the attack, neighboring phones are forced to continuously begin short-lived data connection, thus accessing RACH and flooding it. The authors find out that, for the city of Manhattan, 3Mbps of malicious traffic cause a data and voice connection blocking probability of 65% and, along with that, they point out how attacking data realm could have affect on voice realm too because of the single shared control channel. This fact is extremely interesting and it is important to notice that even outside the data connection realm there are multiple ways to force a mobile phone to access the RACH, thus achieving similar results: the data setup exploited in [27] is just an instance of this effect although it is possibly the one that is most easily kept concealed to the phone owners. Differently from the setup attack, the attack targeting the tear-down mechanism is entirely contained in the data portion of the mobile network, thus it cannot affect the voice network and it can only cause a DoS in the data network. When a new data flow with the user equipment is established, the base station assigns to it a 5-bit Temporary Flow Identifier (TFI) used to mark all packets belonging to the same flow. Once the last packet has been delivered, the base station can release the TFI; however, this event takes place after a 5 seconds delay in order to take into account minor variations in data inter-arrival times. Exploiting this delay a malicious attacker can exhaust all TFIs. A possible example implementation of this attack requires a rogue Internet server answering 32 requests coming from the same neighbourhood with 1-byte-packets sent every 5 seconds. As in the case of the SDDCH attack described before, there is no need for compromised phones.

2.3 Darken the transparent network: attacking Core equipments

A significant advancement in the analysis of mobile network security has been achieved when researchers found a way to attack core network elements, proving that network-wide service deterioration possible. Khan et al. and Kambourakis et al. [18, 17] examine UMTS security architecture finding some protocols flaws that can be used to delete, modify or replay some unauthenticated or not integrity protected messages. This flaws may permit revealing user identities (IMSI), launching DoS attacks against both user phones and network nodes or impersonating the network acting as a man-in-the-middle. These studies, however, do not detail the amount of resources needed to mount a successful DoS attack. An attempt to evaluate the amount of resources needed can be found in the work by Traynor et al.

[26]. The first step is a performance characterization of different HLR devices in different network deployments. The authors identify the transaction most suitable to mount an HLR DoS attack, searching for a compromise between resource consumption and execution time. By means of a simulation of the network behaviour they find that about 11750 infected devices submitting an “insert call forwarding” every 4.7 seconds are sufficient to reduce HLR throughput of legitimate traffic by more than 93%.

Concluding this summary of works related to DoS attacks in mobile cellular networks, it is interesting to notice the “big picture” that [13] and [27] try to draw. Currently studied mobile network DoS attacks roots their cause in the fact that this networks were designed to manage traffic with highly predictable properties but, once connected to the Internet, such constraints hold no more. The Internet was designed with architectural assumptions that are in complete opposition from the ones adopted for cellular networks; this creates a disparity in the effort spent to set up and tear down a connection, necessarily leading to a bottle neck. Moreover mobile terminals have been traditionally considered dumb because of their limited battery life and computational power: this second assumption, however, holds no more in the smartphone era and its underestimation both increases network design complexity and forces core elements to early commit far more resources than those needed by an unauthenticated device. In the following sections we show how it is possible to leverage these facts to greatly reduce the amount of resources needed to mount a successful DoS attack against cellular networks.

Chapter 3

Squeezing radio access protocols

Delivering informations over radio interfaces permits to reach an high number of users without all the costs needed to lay out a physical cable to each costumer's house. This costs reduction, however, comes with more challenging, and thus less efficient, transmission methods than the wired counterpart. This inefficiencies stem mostly from the peculiarities of the transmission medium, but also from the need to grant the main advantage that a wireless communication has compared with a wired one, that is, mobility.

We put ourselves in an attacker's shoes that tries to flood a Public Land Mobile Network (PLMN) with malicious requests asking whether wireless interface is adequate for this purpose or not. In truth, from this point of view, ether may be a good natured bottleneck that force the attacker to deploy an excessive number of compromised devices before reaching his target, thus resulting in a worthless investment.

3.1 Tracking users: the *location update* procedure

Keeping track of the position of every mobile phone, while letting it move arbitrarily inside the area of coverage, is one of the most critical functionality of a PLMN and it is in charge to Mobility Management (MM) procedures. A mobile network is made of cells but always knowing MS position at this level of detail would be both impractical, for the resulting bloat of needed signalling, and useless, because it would require much more MS transmission, even when not in use, thus depleting battery faster. For this reason when the phone is not involved in any communication with the network, the network itself knows its position in a more approximate way represented by two superimposed partitions of the set of cells called *Location Areas (LAs)* and *Routing Areas (RAs).* Location Areas are a concept of circuit switched domain: they are linked with a single MSC/VLR which, in turn, may be responsible for multiple LAs. On the other hand Routing Areas are a packet switching introduction so they are managed by SGSNs instead of MSCs: they are usually smaller than LAs, in order to accommodate the bursty nature of packet traffic, and each RA is fully contained inside only one LAs. To be thorough UMTS introduced another area type, the UTRAN Registration Area (URA), which is not tied with any LA or RA boundary and also allow overlapping between different URAs: this area, however, come into play only in particular conditions out of our interests, so it won't be mentioned further. Being inside a LA/RA couple, the MS informs core network about its actual position using the *location update* procedure. This function requires as an input the user identity represented by the IMSI or, more often, the last assigned (P-)TMSI; as an output, instead, it returns a new (P-)TMSI, meaning that now the MS is *attached* to the network, that is, the latter knows which core elements should be queried to deliver, for example, an incoming phone call. On early deployment of GPRS the location update —also known as attach— procedure should be repeated twice, one time for the Circuit Switching (CS) domain and the other for the Packet Switching (PS); today instead, given the advent of always-connected phones like smartphones, a *combined* function has been introduced: at the cost of a single execution, allows for attach to both domains, thus reducing wait times from power on to the first packet sent.

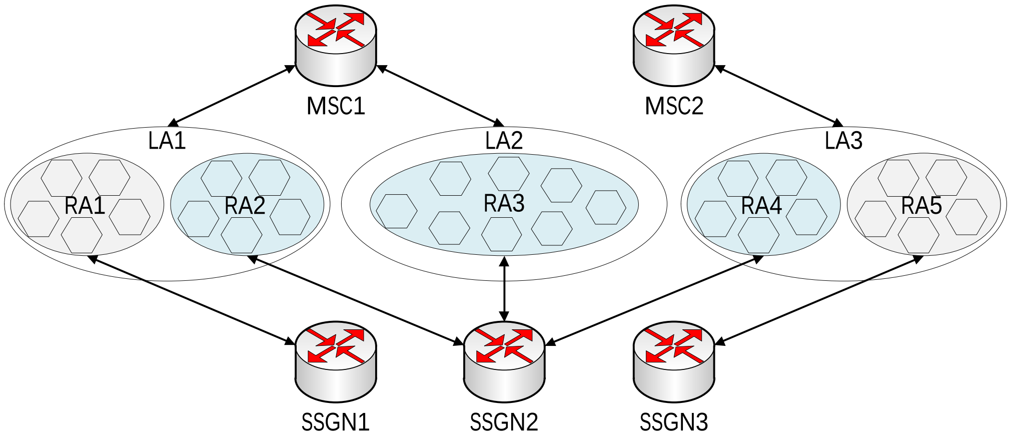


Figure 3.1: Relationship between different area types composing the RAN.

Location updates are triggered when one of the following conditions is met:

- the MS moves from the area code already stored in the SIM card, to another;
- the time elapsed from previous communication has exceeded a configured interval;
- the MS is switched off and on in the same LA/RA: in this case, however, the network may permit a more lightweight procedure called *IMSI attach* which just mark MS' record at VLR active again. The *IMSI attach*, however, often falls back to a full location update execution, so the rest of the thesis will use both terms interchangeably.

Despite differences between GSM and UMTS technologies, that derive from the fact that they use different radio interfaces, a high level description of the attach procedure can be described as follows:

Channel establishment: once MS's modem have scan the air interface choosing, and then synchronizing, with the cell that it considers the best server, the device is ready to make its first request to the network, reporting its location update intentions. This operation is called a *random access* request and it is always carried over a radio resource, also called *channel*, contended with other devices: the network, in fact, still does not know their presence at all, thus cannot allocate dedicated resources for them. Once RAN receives the mobile request the CN usually allocates a signalling channel to carry on successive messages exchange, then delegates back to the RAN the task of making the device actually switch to the dedicated resource. Over the new dedicated channel MS may finally place its location update request sending its identity, usually in the form of the last used TMSI and Location Area, but, when they are not available, the IMSI is used instead.

Authentication and Key Agreement (AKA): before proceeding further in the attach procedure CN may require MS' authentication: this is the case when, for example, IMSI is used as identity declaration. The authentication process begin with MSC asking HLR authentication information for a given IMSI; HLR verify the presence of the IMSI in its database and, aided by AuC, generates a random RAND, which is processed by digest algorithm along with the IMSI's private key $\mathtt{K}_i$ thus obtaining an expected response XRES and a ciphering key $\mathtt{K}_c$. (RAND, XRES, $\mathtt{K}_c$) is the authentication triplet sent back to MSC which, in turn, sends RAND to mobile and receives back SRES as a response: MSC finally claims the user as authentic if and only if XRES = SRES. All the computations on the MS side is performed by the SIM card which is the only other element, apart from HLR, that knows both the digest algorithm and the private key $\mathtt{K}_i$.

MS validation: last product of the authentication phase is the key $\mathtt{K}_c$, which is used from now on for message ciphering between MS and MSC. Inside this protected channel MSC may ask MS to send its IMEI in order to match it against EIR.

(P-)TMSI assignation: being both SIM and equipment valid, the location update procedure concludes assigning a new TMSI, Packet-TMSI (P-TMSI) or both to the MS, depending on the type of attach requested. This is the identifier that will be used for successive communications with the network.

3.2 Vulnerability in location updating

The peculiarity of *location update* procedure is that it cannot leverage any previously accrued knowledge as it must accommodate for new devices of which there is no previous information. Moreover the design described in the introduction, i.e. the model of a smart-network and of dumb terminals, requires the whole procedure to be computationally light for the terminals and to delegate to the network most of the operations and resources. Thus, the terminals do not have to commit significant resources but the network does. These two facts are the basis of the vulnerability to DoS that is present in the attach procedure; in fact, during the AKA step, an unauthenticated device may force the core network to carry on computations that are more resource consuming than the request itself.

As described by Khan et al. work [18], the way an attack could be mounted is straightforward. In a preliminary phase an attacker builds a database of valid IMSIs in a way outlined by [18] itself: whilst there is some commercially-available GSM/UMTS testing and analysing tools that, investing quite a lot of money, may automatize the process, obtaining user identities may also be carried out in a cheaper way, which takes advantage of the opportunity to request IMSI directly from the MS. During the location update procedure, right after MS placed its attach request with TMSI as claimed identity, core network may indeed fail to perform TMSI-to-IMSI translation, for example due to a VLR database malfunction: this circumstance force the network to ask IMSI directly to MS itself. The above mentioned protocol concession happens before any network validation could be made on the MS side, thus allowing an attacker with either a rogue BTS/NodeB, or impersonating the MSC, to coerce MSs into revealing their identities.

The second phase of the attack consists in flooding the network with attach requests each one carrying a different stolen IMSI chosen from built database. The cellular network forwards the requests to HLR/AuC where each IMSI is validated and, being authentic, triggers the calculation of authentication information that are sent back to either MSC or SGSN that, in turn, must submit the challenge back to the mobile station and verify the reply correctness. As the attacker is not controlling the SIM corresponding to the IMSI used, he doesn't know $\mathtt{K}_i$, so he can't calculate the correct answer; however, he does not need to provide it, in fact he does not need to successfully complete the attach procedure, but, on the contrary, his goal is to exhaust HLR/AuC computing resources thus he is already hitting the target with all the valid *attach requests* he is injecting. The second phase outlines also why a list of valid IMSI is indeed necessary: first, using TMSIs requires both the attack and TMSIs' harvesting to be ongoing at the same time because this type of identifiers are ephemeral both in space and time; second, TMSIs force MSC to perform TMSI-to-IMSI resolutions that both dilate execution times and may deplete MSC's resources, causing it to become a bottleneck that reduces attack's effectiveness; finally, a random IMSI may fail HLR validation tests, thus consuming only a minimal amount of resources. Although authors describe this attack with UMTS architecture in mind, it is important to notice that it can be ported, with minimal changes, both to old GSM [14] and new LTE [1] networks.

3.3 Measuring HLR performances

Despite outlining the attack described above, Khan work, however, does not provide a value for the HLR/AuC performance, thus it can't estimate the number of terminals needed by an attacker in order to considerably degrade HLR services. A partial analysis of this problem comes from Traynor et al. article [26]. In this work they outline an attack targeting HLR, but they adopt a different approach that leverages a botnet of authenticated devices, repeatedly injecting resource-demanding transactions available only to already attached terminals. In order to find the transaction that best suits their needs, the authors benchmark the average throughput —in Transactions Per Second (TPS)— of an HLR setup, with respect to different transaction types. Their results are presented in figure 3.2 and point out that the most resource demanding activities are the ones involving both data reading and writing like insertion or deletion of call forwarding rules or the location updating procedure. As a next step authors test on a live network the

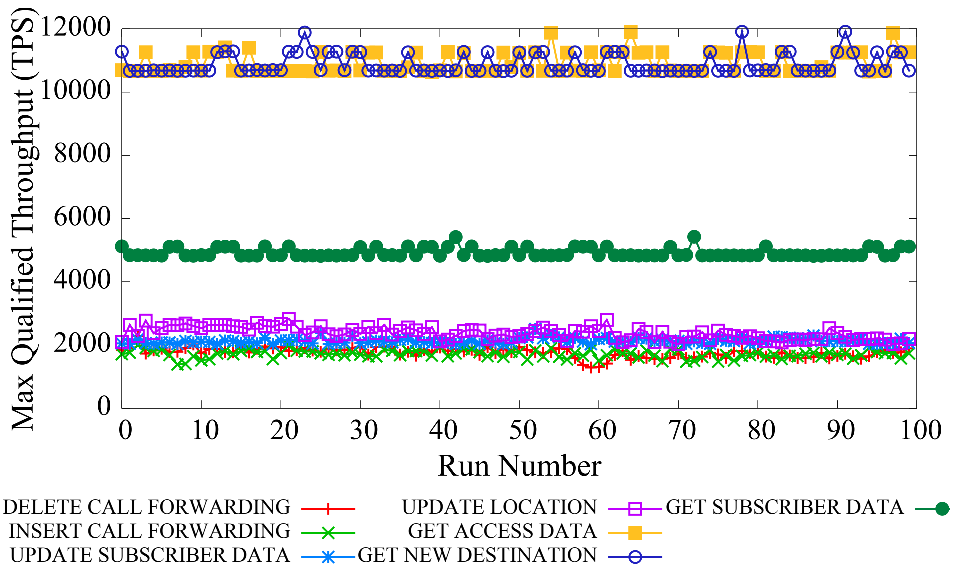


Figure 3.2: HLR throughput for each transaction type in a MySQL setup containing 500k subscribers. [26]

execution time of aforementioned transactions in order to find the best trade-off between computational load and execution speed. They use a mobile phone commanded via *AT interface* [5]: obtained results, presented in table 3.1, lead them to choose the *insert call forwarding* procedure as the attack vector. Table 3.1, however, highlight also a peculiarity of the attach procedure which has been introduced with UMTS: in order to speed up attach procedure and amortize the cost of device authentication SGSN may require more than one authentication triplet[1] where one of them will be consumed on the fly while the others are cached for future use. This is the reason why the authors measure two different timings for the location update both when it hits the HLR and when it just stops at the SGSN utilizing previously calculated challenges.

	Response time
Location update hitting HLR	3s
Location update resolved at SGSN	2.5s
Insert call forwarding	2.7s
Delete call forwarding	2.5s

Table 3.1: Execution times of some HLR transactions measured on a live PLMN. [26]

Authors then simulate the effect of injecting attack traffic on an HLR setup already serving a typical mix of transactions, both in low and high legitimate traffic assumptions. Their simulation results, shown in figure 3.3, permits to determine the rate of malicious requests that an attacker is supposed to deliver in order to achieve a target HLR throughput degradation. Once this rate is defined, equation 3.1 gives the number of needed compromised devices:

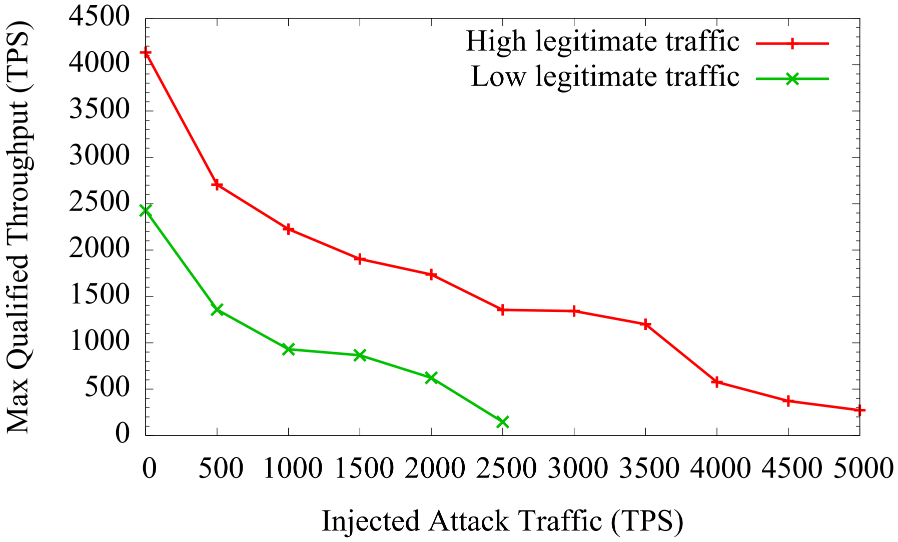


Figure 3.3: Throughput degradation of legitimate traffic on an HLR setup with different attack rates. [26]

$$\text{number of device} = \text{attack traffic (TPS)} \times \text{request period (s)} \qquad (3.1)$$

[1] In UMTS the GSM authentication triplet has been extended with two more information called *integrity key* IK and *authentication token* AUTN which serves respectively for message integrity calculation and network authentication by the MS. Being no more a triplet this five information has been also renamed into *authentication vector*.

Traynor's *request period* equals 4.7s which is composed by 2.7s spent in executing the *insert call forwarding* transaction, whether remaining 2s are a delay guard between successive requests, required by the device. Table 3.2 offers a snapshot of all presented results.[2]

	MySQL setup	
	Low traffic	High traffic
Target TPS degradation	93%	93%
Attack traffic	2500TPS	5000TPS
Request period	4.7s	4.7s
Number of compromised devices	11750	23500

Table 3.2: HLR attack viability based on performance measurements conducted by Traynor et al. [26]

Both figure 3.3 and table 3.2 outline that the more busy the HLR is, the more difficult is disrupting its services. The explanation of this counter-intuitive result resides in HLR equally serving both legitimate and attack requests after reaching its capacity cap. This means that the more legitimate requests are delivered the more their probability of being processed is high or, in other words, only a more powerful attack may convey enough malicious requests so they are more likely to be served instead of legitimate ones.

Traynor's performance measurements in proposed attack. From figure 3.2 it is possible to determine that the *get access data* procedure is roughly 5 times faster than the *insert call forwarding* one, so, in order to achieve the same level of service degradation, we assume that also needed attack traffic must be multiplied by 5. This puts our target to 12500 TPS in low-traffic assumption and 25000 TPS in high-traffic one; however, for the attacker this is a worst case scenario: in fact Traynor's tests focus only on the HLR, disregarding the computations at the AuC that are needed to calculate authentication information.

3.4 Limits of regular mobile phones

To launch the attack Traynor needs a smartphone botnet for two reasons: first, clients must be authenticated before submitting an *insert call forwarding* request; second, this very kind of procedure is a standard one, so it is possible for an application to ask the underlying operating system to begin its execution. In our scenario, instead, regular phones are a limiting factor. First, from a smartphone's

[2] In [26] authors present also a SolidDB HLR setup. However, simulation results regarding this environment are not completely consistent all over the paper; this fact, coupled with an absence of comparison between throughput performance of *insert call forwarding* and *get access data* procedures, lead us to omit this setup in presented results.

OS there's no way to distinguish among the steps of the GSM/UMTS authentication procedure once it has been started: OSes control the modem component via a Radio Interface Layer[3] which converts high level actions such "call number" or "send SMS" into AT commands that the modem logic can understand [5].

Both high level actions and AT commands, however, are too abstract for our needs because the only way to force the attach procedure would be switching the radio off and on again. This operation is completely contained inside the GSM/UMTS protocol stack and operatively hidden inside the baseband module itself, thus the module informs the OS only after the completion or failure of the entire procedure. More in details, in a mobile phone the access to the network can take only one of these three roads:

1. if the device has a valid SIM module, then the attach procedure completes unless there is a failure on the network side;

2. if the device has an invalid —for example expired— SIM module, then it initiate the attach procedure, but the network rejects it without needing a significant amount of resources;

3. if the device has no SIM module at all, then it does not even initiate the attach procedure.

The only way to use a standard phone for performing multiple attach procedures is to equip it with a programmable SIM card and instruct the card to return a different IMSI as well as a random challenge response at each invocation. However, in this case too the solution is definitely sub-optimal because of the phone itself. Built-in mobile protocol stack is implemented strictly following 3GPP specifications which, in turn, are full of transmission wait times, exponential backoffs, maximum re-transmission trials and other artifices [3] designed with the precise purpose to induce a fair use of the network resources. As a proof of this fact Traynor highlights that, during his network behaviour measurements, he was forced to insert a 2s delay between each request: its removal, otherwise, caused extended execution times. The very goal of a DoS attack, on the contrary, is to unfairly squander the network resources in order to prevent legitimate devices to access the service; furthermore we want to reach the limits of the air interface in order to cut down the number of attacking point. For these reasons we claim that the tool best suited to an attacker needs is a dedicated device capable of accessing the network without needing a valid SIM, and without the timing guards and the strict adherence to the protocol that are normally introduced in components aimed at the consumer market.

[3] RIL specifications are available for Windows Mobile® `http://msdn.microsoft.com/en-us/library/aa920475.aspx` (accessed on May 2013) and Android `http://www.kandroid.org/online-pdk/guide/telephony.html` (accessed on May 2013).

3.5 Analysing the Air Interface

We now analyse the peculiarities of GSM and UMTS air interface protocols to evaluate their limits in terms of number of *attach requests* sent to the base station per second. In this process we suppose to be the only device communicating with the target cell; this hypothesis is unrealistic, but is a direct consequence of the unfairness of the attacking device: while legitimate mobile phones would backoff when facing a traffic problem, our device actively works toward the consumption of all the cell's resources. Thus, most of the time a mobile phone tries to get access, it won't be served because of the high number of requests injected by the attacking device, moreover, as soon as a legitimate request completes, the high number of requests injected by the attacking device generates a high probability that the just freed resources will be grabbed by the attacker and made unavailable to legitimate, well behaved devices.

3.5.1 GSM air interface

Um protocol, that is the GSM air interface, has been designed to take advantage of both Frequency Division Multiple Access (FDMA)—like previous 1G technologies—and Time Division Multiple Access (TDMA). Multiple frequencies are mainly used to boost cell capacity in terms of concurrent calls, time division, on the other hand, permits to multiplex multiple voice sources, services and signalling onto the same frequency in order to achieve better spectral efficiency. GSM cells are distinguished one another by having different carrier frequencies that the MS swipes during its boot-up procedures. This particular air resource, being always present, is the one which carries fundamental information for a devices aiming to contact the PLMN along as signalling traffic: for this reason in further explanations we will always refer to this single frequency, focusing instead on the peculiarities of TDMA.

In GSM the atomic part of the time domain is represented by the 8 Time Slots (TSs) composing a TDMA frame. Each TSi s 577μs long and carries what it is called a *burst* of data. There are different burst types to accommodate functions like synchronization, frequency correction, random access and, of course, data delivery: in this latter case the TSt ransport capacity is 114 bits. TDMA frames, whose periodicity is $4.615ms$, are grouped in multiframes which serves two different purposes:

traffic multiframes are composed by 26 frames, thus having a $120ms$ period, and are responsible for voice traffic delivery;

signalling multiframes are made up of 51 frames, thus having a $235.38ms$ period, and deliver signalling and service information.

The complete frame hierarchy presented in figure 3.4 shows two more grouping stages: the superframe acts as an align level for traffic and signalling multiframes; hyperframe's main purpose, instead, is related to communication ciphering. Both this grouping, however, has been cited only for completeness as they won't be cited any further.

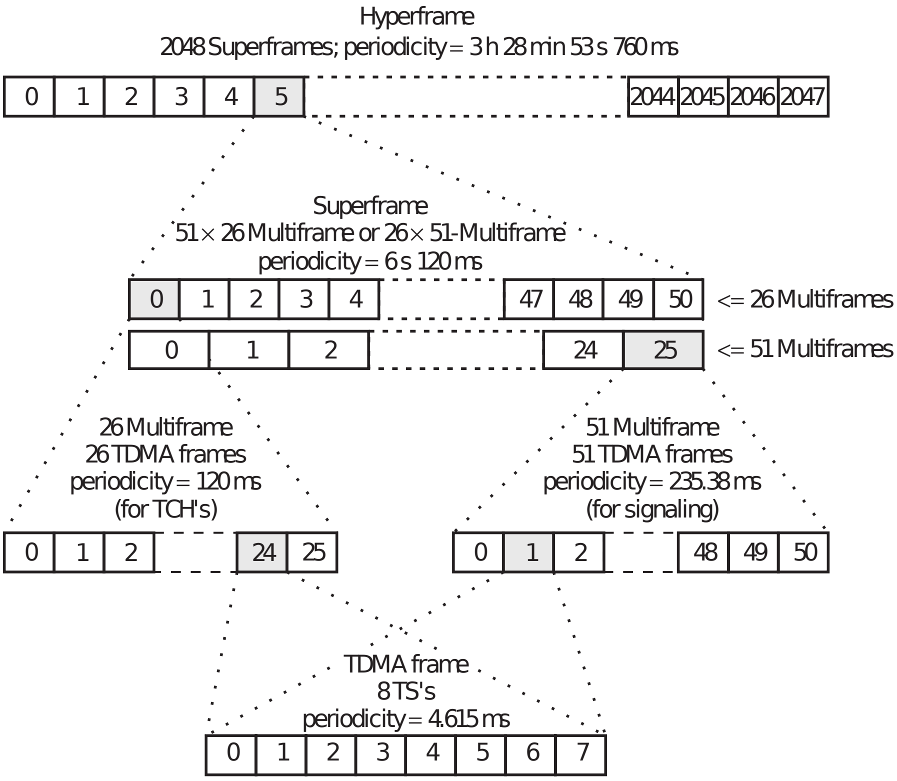
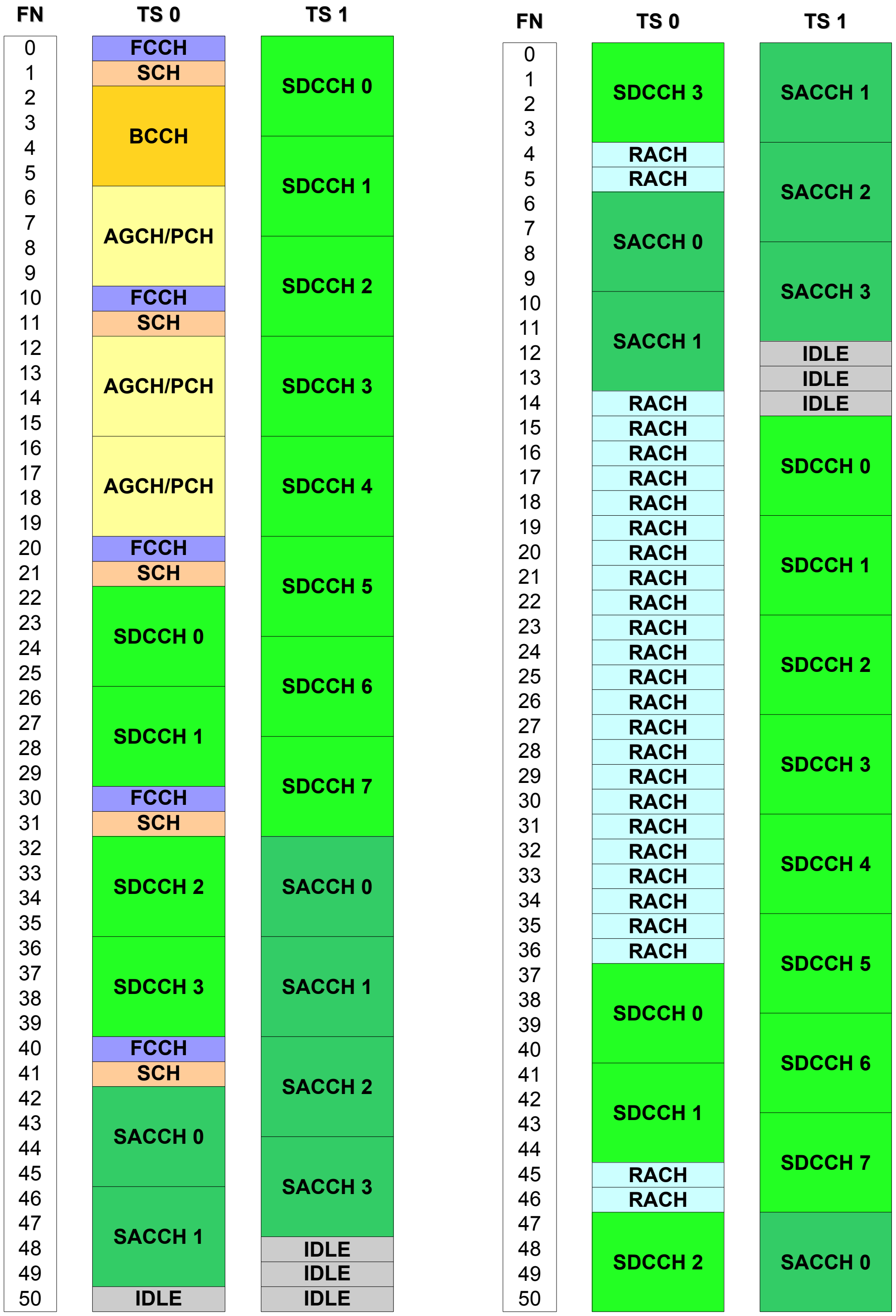


Figure 3.4: Hierarcy of frames in GSM. [14]

The TDMA/FDMA texture lead to the concept of *physical channel* described by a frequency/time slot couple: this means that a single-frequency cell makes only 8 physical channels available. Physical channels are the actual carrier over which different types of data, called *logical channels*, are laid in a time-multiplexed way. For example TS0 of the cell's carrier frequency cyclically transmit signatures for MS synchronization, cell's information, paging signals, etc. spread over an entire signalling multiframe in a preconfigured way. Actually mobile standard dictates the available configurations for signalling multiframes as they mostly vary for the number of available logical channels destined to MS–core network signalling. This particular resource, called SDCCH, is familiar to network planner because it is needed SMSs along as location update or call set-up signalling; being this a critic task the standard allow to compose different multiframe configurations in order to grow or shrink the number of SDCCHs available, thus accommodating different traffic demands. The configuration that we refer to in this thesis is reported in figure 3.5; it commits two entire timeslots for signalling purposes, but also exposes a total amount of 12 SDCCHs, which means that at most 12 MS may use this channel at the same time. [14, 26]

Figure 3.5 shows also that some logical channels occupy 4 successive frames. The reason for this behaviour is that most of the GSM signalling messages are



FN	TS 0	TS 1
0	FCCH	SDCCH 0
1	SCH	
2	BCCH	
3		
4		SDCCH 1
5		
6	AGCH/PCH	
7		
8		SDCCH 2
9		
10	FCCH	
11	SCH	
12	AGCH/PCH	SDCCH 3
13		
14		
15		
16	AGCH/PCH	SDCCH 4
17		
18		
19		
20	FCCH	SDCCH 5
21	SCH	
22	SDCCH 0	
23		
24		SDCCH 6
25		
26	SDCCH 1	
27		
28		SDCCH 7
29		
30	FCCH	
31	SCH	
32	SDCCH 2	SACCH 0
33		
34		
35		
36	SDCCH 3	SACCH 1
37		
38		
39		
40	FCCH	SACCH 2
41	SCH	
42	SACCH 0	
43		
44		SACCH 3
45		
46	SACCH 1	
47		
48		IDLE
49		IDLE
50	IDLE	IDLE

(a) Multiframe configuration in the downlink network segment.

FN	TS 0	TS 1
0	SDCCH 3	SACCH 1
1		
2		
3		
4	RACH	SACCH 2
5	RACH	
6	SACCH 0	
7		
8		SACCH 3
9		
10	SACCH 1	
11		
12		IDLE
13		IDLE
14	RACH	IDLE
15	RACH	SDCCH 0
16	RACH	
17	RACH	
18	RACH	
19	RACH	SDCCH 1
20	RACH	
21	RACH	
22	RACH	
23	RACH	SDCCH 2
24	RACH	
25	RACH	
26	RACH	
27	RACH	SDCCH 3
28	RACH	
29	RACH	
30	RACH	
31	RACH	SDCCH 4
32	RACH	
33	RACH	
34	RACH	
35	RACH	SDCCH 5
36	RACH	
37	SDCCH 0	
38		
39		SDCCH 6
40		
41	SDCCH 1	
42		
43		SDCCH 7
44		
45	RACH	
46	RACH	
47	SDCCH 2	SACCH 0
48		
49		
50		

(b) Multiframe configuration in the uplink network segment.

Figure 3.5: Reference multiframe configuration for the 12 SDCCH instance; missing timeslots are entirely dedicated to voice traffic.

carried over LAPD$_\mathrm{m}$ frames with a static size of 23 bytes; in their way to the air interface this frames are first processed by channel coding algorithms and then spread by interleaver which produced 4 burst suitable to be transmitted by the same number of time slots. Each logical channel, however, is bound to a single physical one, so this 4 bursts shall be transmitted in the same time slot number but in consecutive frames, obtaining this way the occupation pattern depicted in the figure.

GSM attack limits

GSM attach procedure involves only three channels as depicted in figure 3.6: RACH, AGCH and SDCCH. To evaluate the design limits of the GSM protocol we state that it is enough to characterized each logical channel involved both in its multiplicity constraints and utilization time, in order to find out which one introduces the maximum bottleneck. This assumption is backed by the expectation that core network does not pose significant signalling bottlenecks with respect to the air interface of a single cell, moreover, in GSM protocol there is no resources, other than available channels, that may limit the number of user concurrently communicating with the BTS.

Random Access Channel (RACH) analysis

The RACH—the Random Access Channel— is the uplink channel used to carry mobile phone's access requests; in normal conditions, it is governed by the slotted ALOHA protocol, so, in order to maximize its performances, protocol developer designed RACH messages to fill just a single timeslot. We specified "normal conditions" because, in our scenario, we don't care about contention that may be caused by other devices, thus, differently from the normal scenario, we do not apply any backoff and we aim directly at the full channel consumption. In such a scenario, the 12 SDCCHs configuration provides 27 RACH access slots each multiframe and this means a capacity of:

$$\rho_{RACH} = \frac{27}{235.38ms} \approx 114.7 \text{ TPS} \tag{3.2}$$

This result is not fully consistent with the 80TPS calculated by [26] for the slotted ALOHA instance: authors assume a multiframe entirely dedicated to RACH slots, but this is not the case when 12 SDCCHs are deployed, as confirmed by figure 3.5.

Access Grant Channel (AGCH) analysis

The AGCH downlink channel is used to answer incoming random access request; it carries the information needed by the mobile phone to access the dedicated channel used for further communications. Reference configuration allows the BSC to answer up to 3 RACH requests every multiframe by means of the *immediate assignment* commands. The BSC, however, may use also the extended version of this command

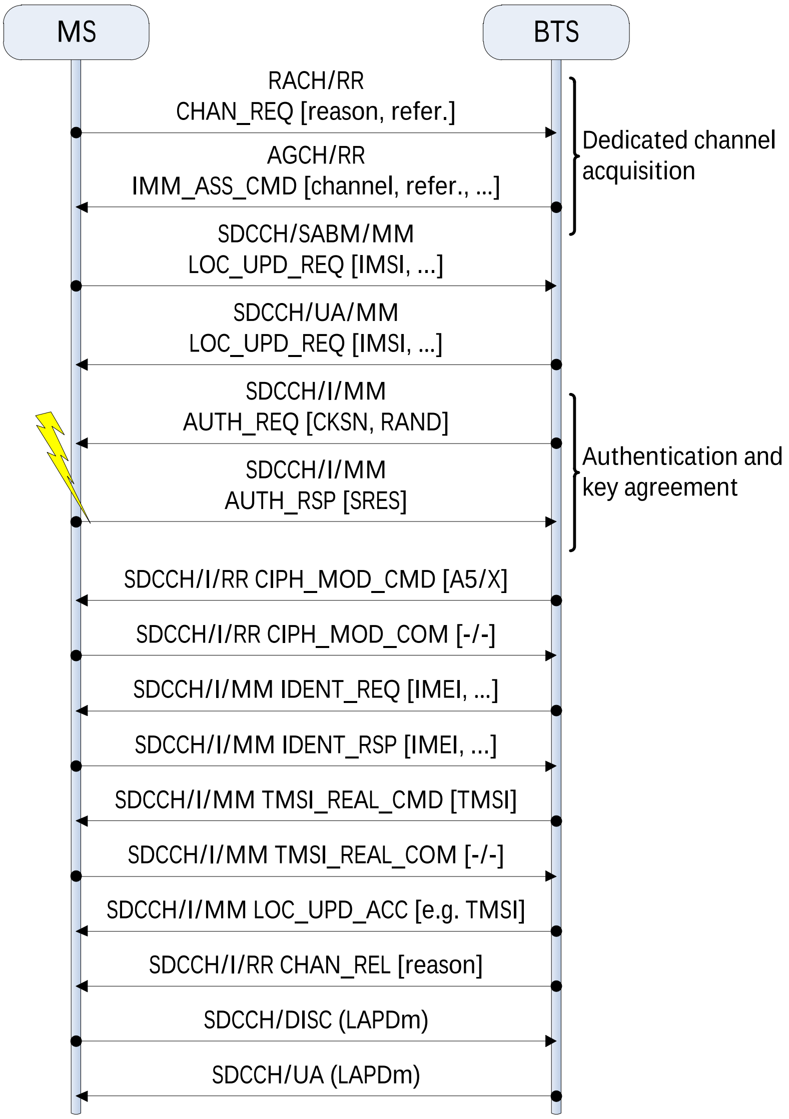

MS
BTS
RACH/RR
CHAN_REQ [reason, refer.]
Dedicated channel acquisition
AGCH/RR
IMM_ASS_CMD [channel, refer., ...]
SDCCH/SABM/MM
LOC_UPD_REQ [IMSI, ...]
SDCCH/UA/MM
LOC_UPD_REQ [IMSI, ...]
SDCCH/I/MM
AUTH_REQ [CKSN, RAND]
Authentication and key agreement
SDCCH/I/MM
AUTH_RSP [SRES]
SDCCH/I/RR CIPH_MOD_CMD [A5/X]
SDCCH/I/RR CIPH_MOD_COM [-/-]
SDCCH/I/MM IDENT_REQ [IMEI, ...]
SDCCH/I/MM IDENT_RSP [IMEI, ...]
SDCCH/I/MM TMSI_REAL_CMD [TMSI]
SDCCH/I/MM TMSI_REAL_COM [-/-]
SDCCH/I/MM LOC_UPD_ACC [e.g. TMSI]
SDCCH/I/RR CHAN_REL [reason]
SDCCH/DISC (LAPDm)
SDCCH/UA (LAPDm)

Figure 3.6: Messages exchanged between MS and BTS during the GSM attach procedure. [14] The lighting on the left marks the message replaced during the attack.

which allows channel assignment to two mobile phones simultaneously, hence doubling AGCH capacity: we will see, however, that even in the non-extended, and therefore more stringent, case the AGCH is not the attack bottleneck. Back to AGCH characterization, it leads to an attack capacity of

$$\rho_{AGCH} = \frac{3}{235.38ms} \approx 12.7 \text{ TPS} \tag{3.3}$$

which indeed represent a tighter limit than RACH.

Standalone Dedicated Control Channel (SDCCH) analysis

The main part of the attach procedure is delivered via SDCCH that is an bidirectional channel assigned to a mobile terminal and reserved to it until a special *channel release* message is issued by the BSC. As we stated above, in our scenario we assume the presence of 12 SDCCHs; determining their occupation time, however, is quite tricky.

Traynor et al. [26] measured an average time of 3s to perform a complete attach where 0.5s are needed by the core network to contact HLR/AuC, calculate the authentication information and receive data back. We prove that the remaining 2.5s are spent to send messages back and forth between the mobile phone and the BTS. A multiframe can carry just one message for each SDCCH in each direction, but, when the BTS requires information to the mobile phone, the latter one can answer in the same multiframe: in fact the GSM protocol states a displacement between downlink and uplink multiframes that allows the MS to compute its reply. Given these two rules and assuming two multiframes needed for the RACH-AGCH exchange, we may conclude that completing the attach procedure requires 11 multiframes, that is $11 \times 235.38ms = 2.6$s that is almost exactly the time obtained in Traynor's measurements. Thus we say that, during message exchange between the MS and the BTS, the only wait time is related to the HLR/AuC interrogation; this, in turn, allows us to estimate SDCCH utilization time during our attack.

The *attach procedure*'s message exchange will be modified during the attack just from *authentication response* message on, in the way depicted in figure 3.7. After receiving the *authentication request* the device answers back with a LAPD_m DISC message that request BTS to terminate the multiple frame operation, releasing its Layer 2 connection [6]. We use this procedure instead of replying with a wrong `SRES` for two reasons: first, it speeds up the SDCCH release cutting the number of needed messages from 10 to 7; second, the *authentication request* message, containing the challenge, already carries the proof that the HLR/AuC has been consulted. Using the same rule, we now require 6 multiframes, 4 of which are carried over SDCCH, leading to a channel holding time of $4 \times 235.38ms + 0.5\text{s} = 1.44\text{s}$, thus a 12 SDCCHs capacity of:

$$\rho_{SDCCHs} = \frac{12}{1.44\text{s}} \approx 8.3 \text{ TPS} \tag{3.4}$$

Comparing each channel capacity and choosing the lower one, we argue that GSM attacking capabilities are limited by the SDCCH channel at a rate of 8 TPS.

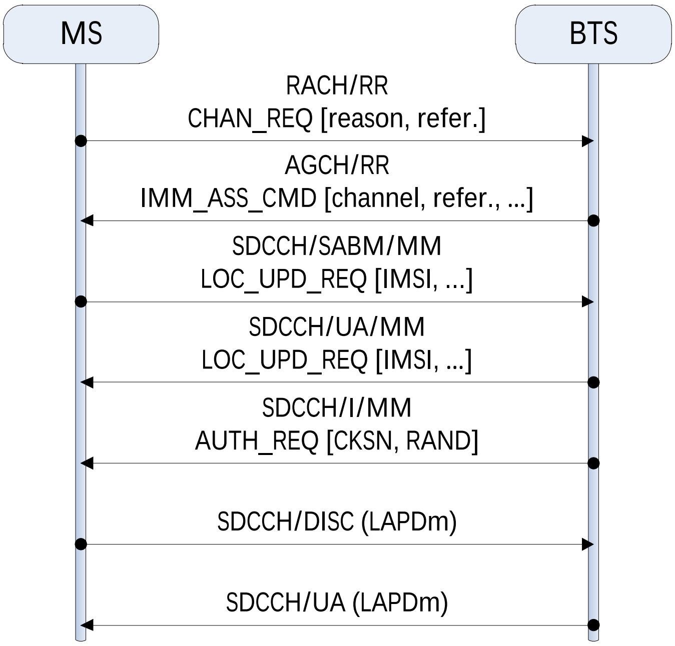


Figure 3.7: Messages exchanged between MS and BTS during the attack: our device solicits an early disconnection right after receiving the `AUTH_REQ` from the network.

This result tells us that in the low legitimate-traffic assumption a GSM-only attack can be mounted with 1563 SIMless devices spread over the same number of cells. This count is already an order of magnitude lower than Traynor's one, but, the multiple RAN architecture makes possible to reduce it even more: for this reason we will now focus on UMTS *location update* procedure, conducting an air interface analysis aimed at finding the limits to its attacking efficiency.

3.5.2 UMTS air interface

UMTS is a mobile cellular system designed to remove GSM inefficiencies related to synchronization between all devices in the RAN. For this reason it substitutes the TDMA protocol with a particular form of Code Division Multiple Access (CDMA), that is Wideband CDMA (W-CDMA), that allows Node B to transmit simultaneously to multiple mobile phones on the same carrier frequency as long as different *channelization codes* are used.

This codes —also known as Walsh–Hadamard sequences— are multiplied with the bit sequence coming out from the channel coding block: the resulting sequence has an higher rate than the input one and UMTS specification fixes it at $3.84M$cps —where the "c" stands for *chip*. Due to the differences in data rates between services, and because the output speed is fixed, the system should be able to apply variant scaling factors. This requisite is feasible because Walsh–Hadamard codes may have different lengths that, once applied to the same initial sequence as in figure 3.8, results in an output rate directly proportional to the code length: this fact leads to the concept of Spreading Factor (SF) which is defined as the number of chips sent for each bit of information.

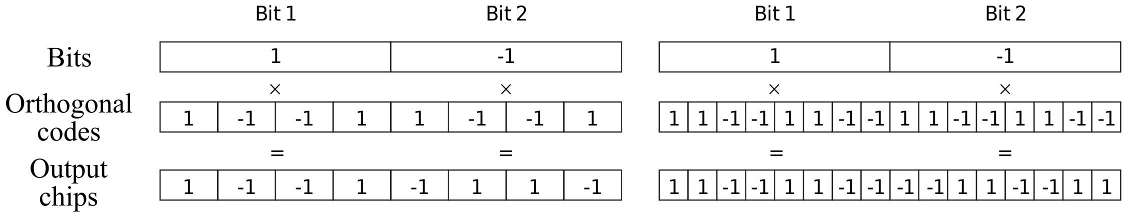


Figure 3.8: Different *spreading* outcomes obtained by multiplying the source signal with Walsh-Hadamard sequences having spreading factor 4 and 8 respectively.

However, the most important property belonging to Walsh–Hadamard codes is *orthogonality* that means that two different sequences of the same length may be multiplied together chip-by-chip and then add up the results leading to a total always equals to zero. In order to obtain orthogonal codes with different lengths the method used is the "binary tree-rule" depicted in figure 3.9 and described by the following recursive equation:

$$H(2^k) = \begin{cases} [1] & \text{if } k = 0 \\ \begin{bmatrix} H(2^{k-1}) & H(2^{k-1}) \\ H(2^{k-1}) & -H(2^{k-1}) \end{bmatrix} & \text{if } k > 0 \end{cases} \tag{3.5}$$

where $H(2^k)$ is a square matrix whose rows are the Walsh–Hadamard codes of length 2^k. This formulation strictly limits the number of available sequences, in fact the number of codes of a certain length equals the length itself; moreover, UMTS documentation bounds code lengths in the range 4–512 further reducing the choice. However not all codes depicted in the figure are mutually orthogonal, orthogonality is indeed respected while choosing among the same-length set, but sequences with a different spreading factor, i.e. different length, are orthogonal so long as they are not ancestors or descendants of each other.

Channelization codes are used in different ways on the uplink and downlink segment of the network: on the downlink portion their purpose is discriminating among different channels which, in turn, may be dedicated to single users; on the uplink segment, instead, orthogonal codes distinguish between multiple connections coming from the same mobile device. This latter point leaves open question: how the Node B may recognise a mobile station from the others? This task is accomplished by *scrambling* codes which also serves to distinguish different Node B signals: all UMTS' Node Bs actually transmit on the same frequency range so, this is the mean by which MSs can selectively listen to them. Scrambling codes are multiplied with the signal after spreading codes but, being 38400 chips long with a repetition of $10ms$, this time resulting rate is not changed. Their generation

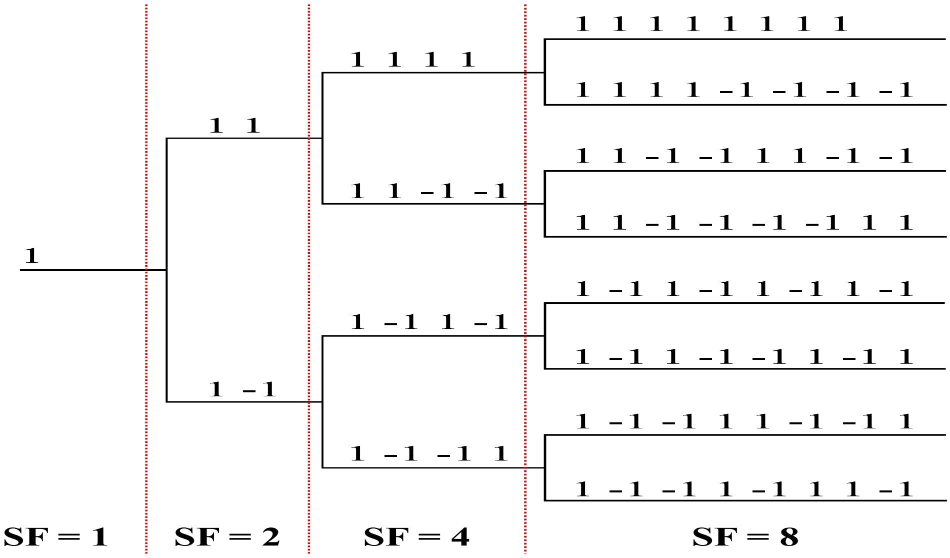


Figure 3.9: Portion of the spreading codes tree: UMTS uses lengths in the range 4–512.

process uses a pseudo-random number generator which makes this codes *uncorrelated*. This characteristic is looser than orthogonality, it results in a much higher number of available codes —about four million— but also cause a certain amount of interference between signals because the multiplication of two scrambling codes bit-by-bit and then summing up obtained results gives a total that is zero only on the average: this obviously leads to an higher and higher chance of receiver errors as new devices joins the network.

UMTS attack limits

The complete UMTS *location update* procedure is very similar in its phases to the one already presented for GSM (Fig. 3.6), for this reason here we focus only the message exchanges between MS and Node B during the attack, illustrated in figure 3.10.

The first message that deviates from a standard *location update* flow is the same as in GSM, that is, the authentication response message. Unlike GSM, however, this time the attacker has to reply to the authentication request with a wrong challenge response `SRES` because, at this stage, the UMTS protocol stack does not allow a MS-initiated connection release: neither at RRC layer [2], nor at RLC one [4]. The attack vector of figure 3.10 is exactly the same described by Khan et al. [18] and it is the one that uses as few Node B/SGSN resources as possible in order to not make the processing power of this devices an unintentional bottleneck. This solicitude is right in the opposite direction of the attack proposed by Kambourakis et al. [17] that aims to overstress both HLR and SGSN: authors modify the MS capabilities declared in the initial *GPRS attach* message; doing so the *location update* procedure execute flawlessly until the *security mode* command

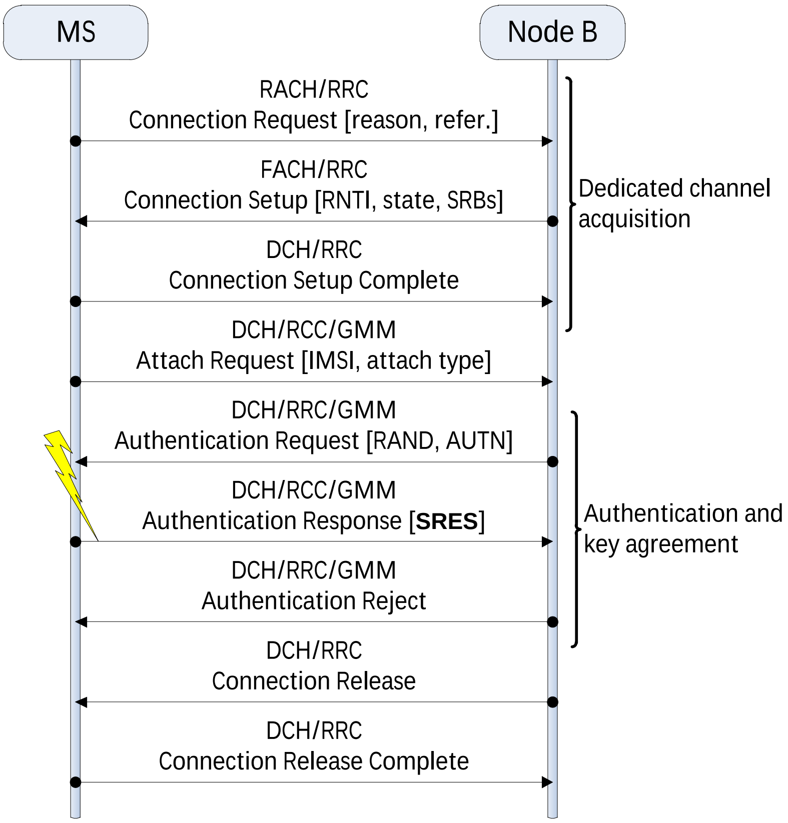

MS
Node B
RACH/RRC
Connection Request [reason, refer.]
FACH/RRC
Connection Setup [RNTI, state, SRBs]
DCH/RRC
Connection Setup Complete
DCH/RCC/GMM
Attach Request [IMSI, attach type]
DCH/RRC/GMM
Authentication Request [RAND, AUTN]
DCH/RCC/GMM
Authentication Response [SRES]
DCH/RRC/GMM
Authentication Reject
DCH/RRC
Connection Release
DCH/RRC
Connection Release Complete
Dedicated channel
acquisition
Authentication and
key agreement

Figure 3.10: Messages exchanged between MS and Node B during the attack. The lighting on the left marks the message carrying the wrong `SRES` response.

is issued, that is, when the MS checks the security capabilities early received by the network and, noting the inconsistency, terminates the procedure. We argue, however, that trying to obtain also an SGSN DoS requires a much more careful attack design because, otherwise, this device may easily become an obstruction for request targeted at the more capable HLR.[4]

Before continuing to an in-depth analysis of the air interface we should estimate how long dedicated resources are kept occupied by a single request. To this end Johnson et al. [16] profile the delay time of an UMTS data connection setup, that is, the elapsed time from initial *rrcConnectionRequest*, after radio powers up, to the first UDP packet sent. Before sending an UDP packet the MS should establish a Packet Data Protocol (PDP) context which, in turn, requires the device to be located and authenticated, that is, MS should perform a complete *location update* run: here's why the article proved to be valuable. Authors' analysis covers both a Signalling Radio Bearer (SRB) capability of 3.7$k$bps and 14.8$k$bps. An UMTS Radio Bearer is a data streams that spans multiple network elements with a defined Quality of Service (QoS), bitrate, acknowledgement mode and other parameters defined both by documentations and network planners. Radio Bearers allocated for signalling are typically declined with the two bitrates stated above: the 3.7$k$bps is the most common of the two because it uses less resources, but, when signalling traffic gets higher, the switch to the more capable and more resource-expensive 14.8$k$bps SRB may be necessary.

Table 3.3 shows that the MS receives the *security mode* command at 1160ms and 850ms respectively: this message is what a mobile usually receives after it pass the authentication phase. In our scenario it will be replaced by the *authentication reject* dispatch, followed 10m later by the *rrcConnectionRelease*: this 10ms delay is due to the Transmission Time Interval (TTI) of the UMTS signalling frame and supposes a channel without jitters.[5] Lastly, 10ms after the request of connection release, MS replies with a *rrcConnectionReleaseComplete* roughly at 1180ms for the 3.7$k$bps case and 870ms for the other. This values, however, does not include HLR/AuC interrogation overhead that authors estimate of about 600ms, thus resulting in a total procedure time of 1780ms and 1470ms for the 3.7$k$bps and 14.8$k$bps respectively. We want to notice that these timing may be overestimated in our scenario, because the *security mode* command force network elements to activate ciphering and integrity protections routines: this overhead is obviously not present when the authentication request is rejected.

The high-level description of UMTS already defined two of the three constraints that limit the number of users a Node B may concurrently service: channelization

[4] [17] proposes also to couple its attack with a database of stolen IMSI in order to cause a much more serious damage: this is however impossible because the attacker, without knowing $\texttt{K}_i$, would not pass the authentication phase and hence, would not reach the *security mode* command needed to trigger the additional resource depletion.

[5] Obviously this assumption does not hold in real world examples, but, being the attacker able to place the devices wherever he wants, we may assume that differences in inter-arrival times can be limited enough to be ignored or amortized by other approximations.

Table 3.3: UMTS *location update* setup dalays, as registered by a MS. [16]

	3.7$k$bps SRB	14.8$k$bps SRB
RRC Connection Request	0ms	0ms
RRC Connection Setup	390ms	400ms
RRC Connection Setup Complete	590ms	590ms
GPRS Attach Request	590ms	600ms
Security Mode Command	1160ms	850ms
Security Mode Complete	1160ms	860ms
GPRS Attach Accept	1560ms	1040ms

codes and interference; the third one that remains to be mentioned is network access. We will now proceed in the analysis of all this aspects to identify the most stringent one in terms of attacking capacity.

Random Access analysis

The first UMTS bottleneck we are going to estimate is random access. Before accessing RACH the MS has to send out some short preambles, with increasing power, until Node B acknowledge their reception over Acquisition Indicator Channel (AICH): the procedure is defined this way in order to select the minimum power needed to reach the Node B itself. Preambles consists of 256 repetitions of a 16 chips long Hadamard sequence so, there are 16 sequences the mobile may randomly choose from. Once the output power has been calibrated the mobile phone may transmit its single transport block message over RACH, which usually takes a 20ms transmission time interval. Sticking with the single-device hypothesis already used for GSM, and stating that, given the attacking device stationary, it takes just one preamble to demand Node B attention, we estimate a total RACH utilization time of 30ms. This sentence, however, involves also the assumption that the *rrcConnectionSetupComplete* message is not sent over RACH: we state that this is the case because MS early declares its attach intentions in the *rrcConncectionRequest*, so the network prefers to redirect the high amount of successive signalling traffic over a Dedicate Channel (DCH), instead of polluting the shared one. This basis, coupled with AICH capability to acknowledge up to 16 preamble signature at the same time, lead to a random access capacity of:

$$\rho_{PRACH} = \frac{16}{30ms} \approx 533 \text{ TPS} \tag{3.6}$$

Forward Access Channel (FACH) analysis

Once the network received the *rrcConnectionRequest* it assigns dedicated resources via *rrcConnectionSetup* message sent over FACH, a shared downlink channel. This message is relatively large as it typically requires seven transport block of 168 bit each, transmitted, multiplexed in couples, using 10ms TTI [16]. This led to a total capacity of FACH channel of:

$$\rho_{FACH} = \frac{1}{7/2 \times 10ms} \approx 28.5 \text{ TPS} \tag{3.7}$$

which is also consistent with FACH throughput of 32–33$k$bps commonly used in literature.

Downlink network segment analysis

When RRC-layer connection has been established further message exchanges are carried on a per-user dedicated channel. This means that on the downlink segment the number of simultaneous user is limited only by cell transmitting power and the number of available channelization codes. Transmitting power, however, does not pose major hurdles giving that the attacking device will be placed near the antenna and never moves. On the other hand we already see that channelization codes are a scarce resource but, in order to estimate the number of available ones, we have to conjecture about the spreading factor used by dedicated channels. Given that uplink throughput is usually lower than downlink one, we use user-layer uplink DCH data rates calculated in [15] to identify sufficient SFs; dedicated channels, however, have to share available codes also with common channels and this represents an overhead of about 10 codes with SF = 128. [15] We are now able to derive downlink channel capacity using the timing assumptions already described above: while given values are comprehensive of the access phase over RACH/FACH, we need to keep it included because when MS receives the *rrcConnectionSetup* message its dedicated channel has been already reserved. Results taking into account all these factors are presented in table 3.4.

Table 3.4: Downlink attacking capacity calculations.

	3.7$k$bps SRB	14.8$k$bps SRB
Spreading Factor	256	128
Available dedicated channels	236	118
Channel occupation time (s)	1.78	1.47
$\rho_{\text{DLchannels}}$ (TPS)	132.6	80.3

Uplink network segment analysis

The uplink segment uses scrambling codes to distinguish between transmissions coming from different MS. This codes, however, causes interference with each other, thus it is not possible to arbitrarily add new mobile stations to the system, trying to exhaust all available scrambling codes: for this reason CDMA networks are referred to as being *interference-limited* systems. The estimation of the number of device that may concurrently access the air interface is subordinated to two concepts: *pole capacity* and *Rise Over Thermal (ROT).* Pole capacity is the theoretical maximum capacity of the system due to interference. Under the hypothesis of perfect power control, where all devices are received with the same power, and quasi-orthogonal codes, that scrambling codes approximates, pole capacity can be written as:

$$\text{Pole Capacity} = \frac{W}{R_b} \times \left(\frac{E_b}{N_0}\right)^{-1} \tag{3.8}$$

where W is the chip rate fixed, in W-CDMA, to $3.84M$cps, R_b is the user data bit rate and E_b/N_0 strictly speaking, is the energy per bit to noise power spectral density ratio: in order to estimate its value, it has to be taken into account transmission characteristics like receiver sensitivity, channel description, modulation and channel coding types etc. For our calculations we considered an $E_b/N_0 = 6dB$ which is $1.5dB$ higher than the state of the art estimation for a voice uplink "pedestrian" channel presented in [15].

Pole capacity, however, is just a theoretical limit because the uplink noise rise as $(1-\eta)^{-1}$, with η giving the cell load factor; this means that when η approaches 1 also power needed to keep the same E_b/N_0 at receiver moves toward infinity. [22] This phenomenon is called Rise Over Thermal and force the system to work away from its analytical limit: typical configurations account for a maximum load factor of $\eta = 75\%$. [15] Composing presented constraints we are able to measure capacity on the uplink channel which numerical results are presented in table 3.5.

Table 3.5: Uplink attacking capacity calculations.

	$3.7k$bps SRB	$14.8k$bps SRB
E_b/N_0	$6dB$	$6dB$
Pole capacity	260	65
η	75%	75%
ROT capacity	195	48
Channel occupation time (s)	1.78	1.47
$\rho_{\text{ULchannels}}$ (TPS)	109.6	32.7

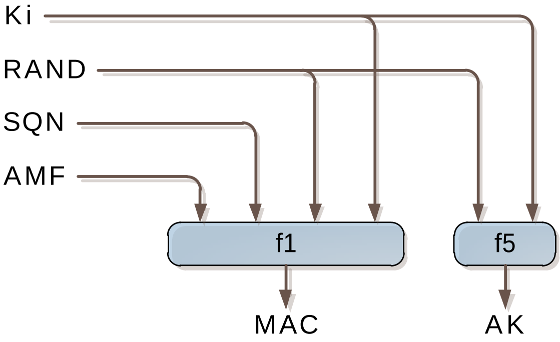
During uplink capacity calculations we have to pay attention that ROT capacity does not exceed the number of available downlink dedicated channels, indeed, the comparison between tables 3.4 and 3.5 confirms found results. Another interesting side note to uplink and downlink calculations concerns the higher attacking capacity of the $3.7k$bps SRB with respect to the $14.8k$bps one. The $3.7k$bps SRB is indeed slower in performing *location update* signalling than the other but tables 3.4 and 3.5 show that the latter has lower "attacking efficiency" because it requires a lot of resources that do not match with the modest procedure speed-up.

The comparison of bottlenecks found so far shows that the hard limit of UMTS attacking capacity is given by the FACH channel at a rate of roughly 28 TPS. The fact that this limit is given by a channel used just to carry a single message, instead of dedicated ones, may be explained by the fact that UMTS system's design has privileged throughput maximization for high-load, long-standing connections: for this reason a connection setup requiring a bit more resources than in GSM is well amortized in following exchanges. Our attack requests, on the other hand, are fast and bandwidth-limited, that is, right in the opposite direction from typical UMTS transaction and, as a direct consequence, they clash with the increased connection setup complexity.

The attack rate found above, however, is not the absolute limit achievable via *attach procedure*, but, in order to push it to full capacity, each device has to know the IMSI' secret keys $\texttt{K}_i$ that is, we have to remove the SIM-less constraint.

Doubling UMTS attacking capacity using SIMs

Given the peculiar limitation discovered so far, we deeper investigated UMTS specification that covers the attach procedure [2], looking for any stratagem that would force the core network to query the HLR more than once before tearing down the ongoing signalling connection. Luckily enough the protocol allows this kind of trick. With respect to GSM, UMTS security has been improved under many aspects and some of them, for example network authenticity check, even represent new entries over previous generation. Testing the authenticity of the network allows a MS to disclose an attacker trying to impersonate the network itself with, for example, a rogue Node B. The key information needed in the process is the `AUTN` value sent with the *authentication request* message and obtained as described in figure 3.11. This generation employs a random value `RAND` which state output freshness, the Authentication and Key Management Field `AMF` that contains some informations regarding MS's network validation tolerance and key lifetime and, especially, IMSI's secret key $\texttt{K}_i$ and a particular sequence value `SQN` which is incremented after each successful authentication: these last two information are kept strictly secret by MNOs thus only a legitimate network that knows both of them could create a valid `AUTN`.



$$\text{AUTN} = \text{SQN} \oplus \text{AK} \parallel \text{AMF} \parallel \text{MAC}$$

Figure 3.11: Information involved in calculating `AUTN` value.

The MS may incur in different failures during the `AUTN` check; one of them regards the `SQN` value being out of the correct range, which in turn lead MS to inform the network about detected problem with an *authentication failure* message, reporting *synchronization failure* as justification. Upon receiving this error message, the SGSN should perform the *re-synchronization* procedure:

1. delete all unused authentication vectors for the faulty IMSI;

2. obtain new vectors from the HLR, based on information attached to *authentication failure* message;

3. initiate a new authentication procedure sending the MS an *authentication request* with one of the freshly obtained authentication vectors.

This process, however, may be executed just once because 3GPP documentation [2] explicitly states that the network may terminate the authentication procedure if two consecutive *authentication failure* messages are received.

The way an attacker may take advantage of this protocol allowance is straightforward as reported by message exchanges in figure 3.12. Despite the attack simplicity the figure specify that the *authentication failure* message carries with the justification also the `AUTS` value. `AUTS` contains information used by the network to prepare the fresh set of authentication vectors but, the critical point for the attacker is that it cannot be spoofed, thus requiring valid SIM cards. Picture 3.13 explains how `AUTS` is calculated and shows that, as long as requisites of functions $f1$, $f1^*$, $f5$ and $f5^*$ holds, it is robust against following threats:

replay attack: the `RAND` value is the same used by the network to compute `AUTN` so it states the freshness of received `AUTS`;

eavesdropping: the value contained by `AUTS` —that is MS' $\mathtt{SQN}_{\mathrm{MS}}$— is concealed using both $\mathtt{K}_i$ and `RAND`;

tampering: `AUTS` is authenticated using IMSI's private key.

The attack capabilities of this modified version of the *location update* procedure can be derived quite easily from previous calculations. Leaving aside random access, which does not pose any limitation even in the standard attack way, we state that current FACH capacity doubles the old one. The reason is that for each RRC connection set up, now the attacker is able to query the HLR twice, hence resulting in this channel carrying up to

$$\rho'_{FACH} = \frac{2}{7/2 \times 10ms} \approx 57.1 \text{ TPS} \tag{3.9}$$

Before declaring this result as conclusive we should check that timing extension due to the second HLR interrogation does not cause downlink and uplink segments to become the new bottlenecks. Comparing two message exchanges of figure 3.10 and 3.12 we note that, with respect to the standard attack, the "synch failure" one just requires another full authentication phase plus the *authentication failure* message:

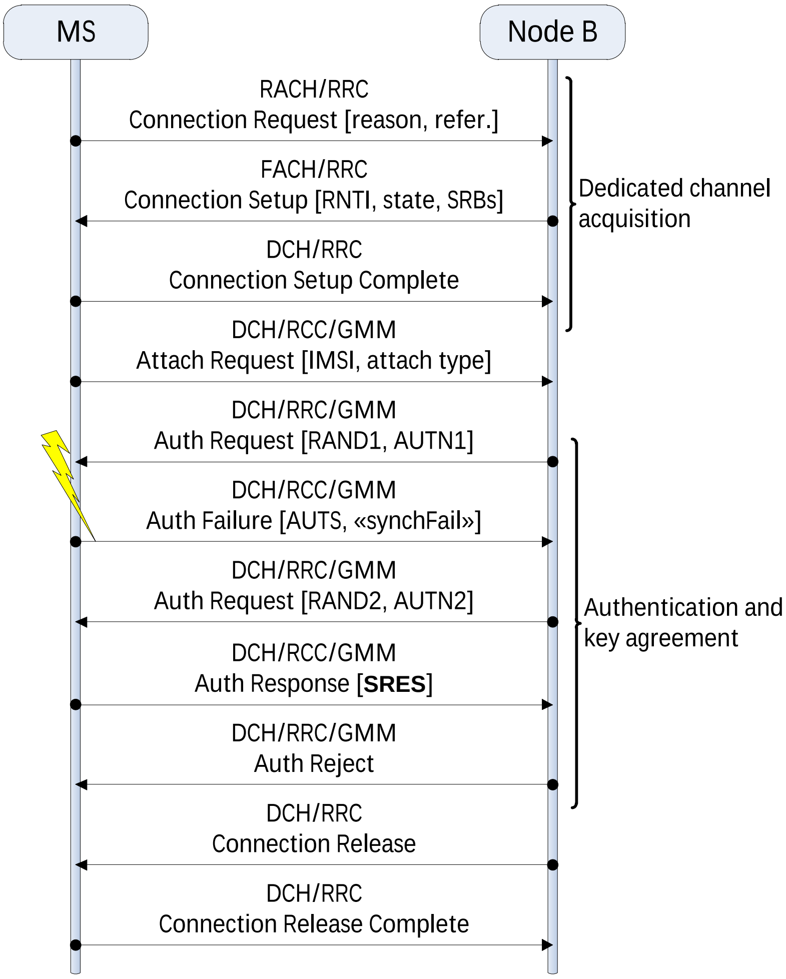
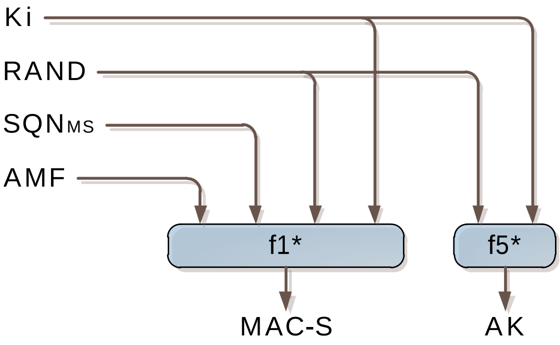

MS
Node B
RACH/RRC
Connection Request [reason, refer.]
FACH/RRC
Connection Setup [RNTI, state, SRBs]
DCH/RRC
Connection Setup Complete
DCH/RCC/GMM
Attach Request [IMSI, attach type]
DCH/RRC/GMM
Auth Request [RAND1, AUTN1]
DCH/RCC/GMM
Auth Failure [AUTS, «synchFail»]
DCH/RRC/GMM
Auth Request [RAND2, AUTN2]
DCH/RCC/GMM
Auth Response [SRES]
DCH/RRC/GMM
Auth Reject
DCH/RRC
Connection Release
DCH/RRC
Connection Release Complete
Dedicated channel acquisition
Authentication and key agreement

Figure 3.12: Messages exchanged between MS and Node B during an attach attack that uses the *synchronization failure* strategy.



$$\text{AUTS} = \text{SQN}_{\text{MS}} \oplus \text{AK} \parallel \text{MAC-S}$$

Figure 3.13: Information involved in calculating `AUTS` value.

we already see that the former takes about $600ms$; for the latter we estimate an execution time of about $100ms$ which represents a conservative average of message delivery timings profiled by Johnson et al. [16] This assumptions lead to a total execution time for the "synch failure" attack of 2.48s[6] then, focusing on the uplink network segment, which we proved to be the most constrained, its capacity become

$$\rho'_{\mathrm{ULchannels}} = 2 \times \frac{195}{2.48\mathrm{s}} \approx 157.2 \text{ TPS} \tag{3.10}$$

that still represents an improvement over the 109 TPS of standard attack. Indeed this is an expected result because, while the number of HLR interrogation has doubled, the resource occupation time only increased by nearly 40%, therefore resulting in higher efficiency also for dedicated channels.

This UMTS analysis shows that an attacker targeting the *location update* procedure of the UMTS protocol may inject up to 28 TPS to the HLR, thus being able to cause major service degradations with as few as 446 SIMless devices in the low legitimate traffic assumption and 892 devices in the high legitimate traffic one. However, in order to grasp the full potential of the *location update* procedure the attacker may permit SIMs use, therefore doubling the number of requests sent each second, which results in only half devices needed with respect to the SIMless case.

3.6 Composing GSM and UMTS attacks

We have proved that causing major service deterioration using SIMless devices is not only possible but, compared with the number of devices required for a botnet based attack, allows reducing the amount of resources by more than an order of magnitude. This result is achieved exploiting a single radio technology at a time but our attacking device can be designed to deal with both GSM and UMTS at the same time, therefore being able to compose their attack capacities. We can't concretely prove these claims because said device has not been built yet, but we now show why current technology allows us to claim that it is possible to compose GSM and UMTS capacities in the way that is most profitable from the attacker point of view, i.e. the two technologies capabilities may be simply summed up.

3.6.1 Physical device feasibility

For what concerns technology GSM and UMTS over the air interface use different frequency bands and this means that the device should be equipped with two analog radio frequency modules, and a couple of baseband processors with enough processing power to be able to keep track of all concurrent communications. Multiple analog modules are a standard equipment of every modern mobile phone destined to the medium or high-end market: it simply processes signals without

[6] We refer to the $3.7k$bps SRB case only, because we already pointed out how the $14.8k$bps SRB proved to be less efficient.

knowledge of what it is carrying and it's already available on the market. Baseband processor, instead, is a critical part because it has to deal with as many different bit stream as the number of ongoing HLR requests. The proof of its feasibility is the very presence of Node Bs and BTSs that are able to handle all the traffic but, we speculate that this is an upper bound in complexity and that the actual device requires a significantly simpler design as it does not need to process all the possible events in the complete protocol or high bandwidth demands.

The sketch in figure 3.14 is a prototype of the attacking device, that employs one baseband processor for each concurrent request which, in turn, is connected to a multiplexer/demultiplexer (mux/demux) that compose all incoming bit streams to produce a single output signal to be sent to the analog module. This is surely an inefficient design, because low bandwidth requirements of signalling channels do not allow an efficient exploitation of the whole processing power, however it represents a reference in what follows.

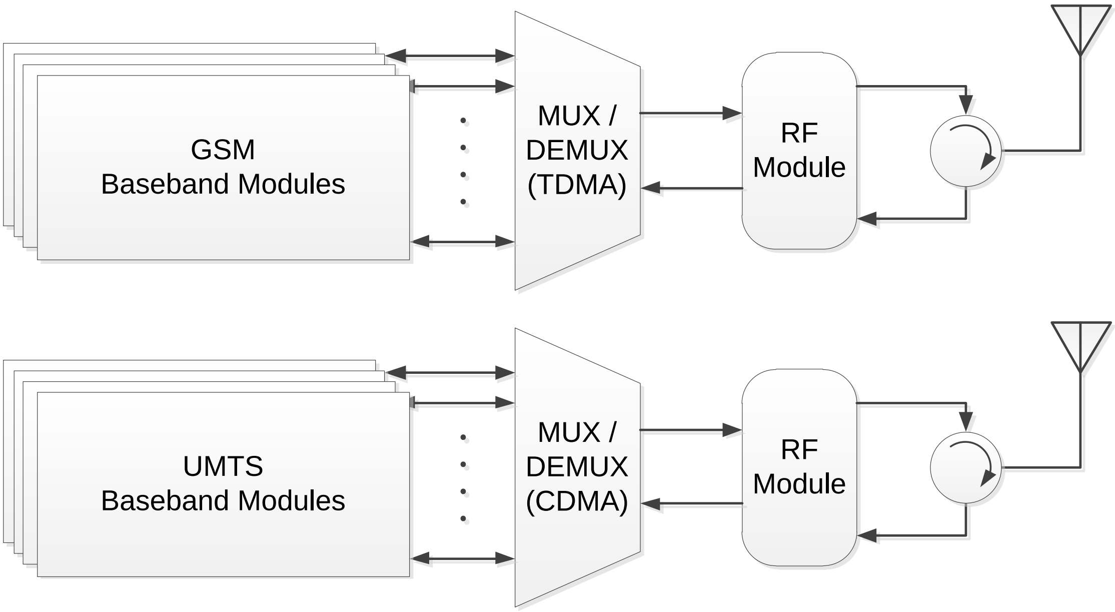


Figure 3.14: A prototypical sketch of the attacking device's functional parts.

For both systems only a small circumscribed part of the protocol should be implemented and some functions, like the composition of physical channels, should be moved from the baseband module into the MUX/DEMUX component. Moreover there is no need to waste processing power on auxiliary functions like handover because, being the device static, neighbouring cells received power can be computed once and returned whenever asked. In GSM system the device is tuned on the carrier frequency and requires just a time (de)multiplexer to perform correct (de-)channelization. Transmissions and receptions, however, happen only on TS0 and TS1 of every frame: this is indeed twice the load a mobile phone is subjected to during a call, but, being the raw processing power more than doubled from first

GSM-phone introduction, it should not pose any problem. Moreover transmitting in two timeslots means that the temporal disjunction between talking and listening periods is no longer valid: this problem can be addressed with a duplexer. UMTS system, on the other hand, presents the main challenges for what concerns processing power because all devices may talk simultaneously and also the (de)multiplexer part is no more on the time domain, but has to account for all possible scrambling and orthogonal codes uses by each channel of each connection.

3.6.2 Network load separation

The transactions carried over GSM and UMTS hit BTSs and Node Bs respectively that are usually different pieces of hardware with their own processing power. Moreover we already described that the *location update* procedure may be asked for the circuit and packet switched domains separately, that is an attacker can force BSCs and RNCs to deliver packets either to MSC or SGSN in a mutually exclusive way. Using these two considerations along with an adequate dispersion of attacking devices we suppose that the attacker is able to avoid network bottlenecks balancing the load on different equipments. Certainly this is just a first level description of the network separation problem, however, a thoroughly one would require accurate modelling and analysis of all the backhaul protocols and such a task is outside the scope of this thesis..

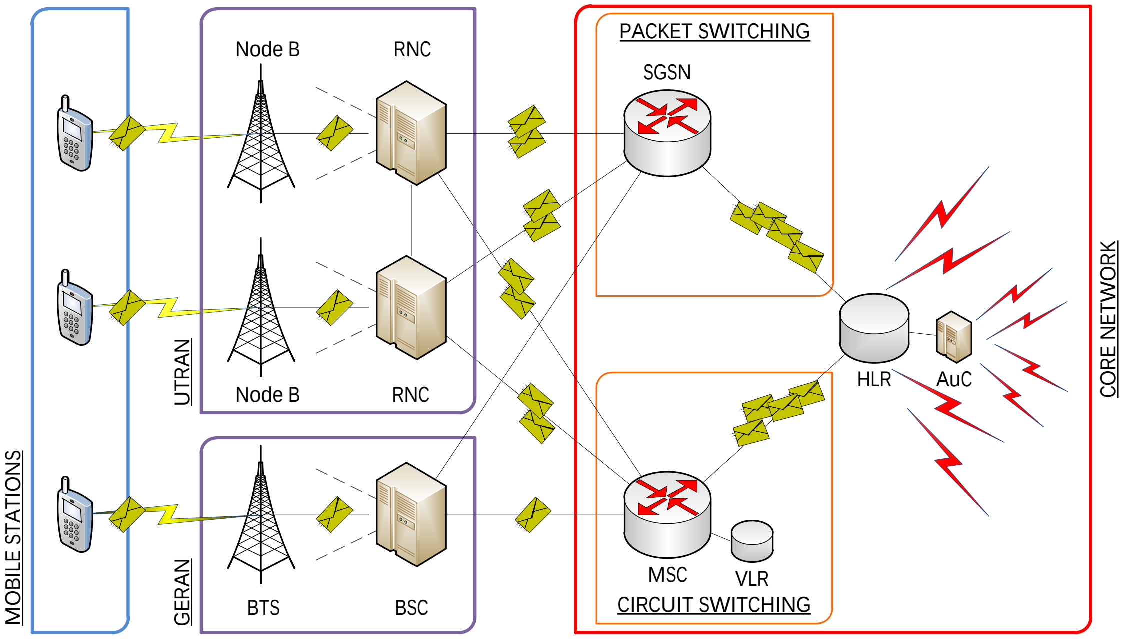


Figure 3.15: Using packet or circuit switched *location updates* on GSM or UMTS systems it is possible to affect different core network elements thus balancing traffic flows.

3.6.3 Summing capacities

The descriptions provided in the previous sections show that it is possible to compose the attacking capabilities of each RAN in a purely additive way, thus each device capable of exploiting different RANs delivers a significantly increased attacking capability. Results found so far, and summarized in table 3.6, show that with 347 SIM-less devices or as low as 192 SIM-equipped ones, it's possible to inject up to 12500 *location update* requests each second, aimed at depleting HLR computing resources.

	SIM-less device		SIM-equipped device	
GSM attack rate	8 TPS		8 TPS	
UMTS attack rate	28 TPS		57 TPS	
Total attack rate	36 TPS		65 TPS	
	Low traffic	High Traffic	Low traffic	High Traffic
Target attack rate	12500	25000	12500	25000
Needed devices	347	694	192	384

Table 3.6: Summary of the attack rates deliverable via the attach procedure.

Chapter 4

Conclusions

Cellular networks are one of the infrastructure designated as critical both in the American and the European vision of the homeland security. This has lead to a large number of studies that have analysed the architecture of the networks to identify and possibly mend vulnerabilities that could be exploited to mount attacks.

Each infrastructure has been deeply analysed and many possible sweet spots for an attack have been neutralized; however, two new factors aggravate the complexity in the infrastructure defence. The first of these factors is the appearance of programmable mobile phones; the second aggravating factor is, as it has been already pinpointed in previous works [7], the interplay between different well known components: in this case coexisting different generations of networks. In past works several ways to mount DoS attack, leveraging the programmability of modern smartphones, have been described, however, these works characterize methodologies that needed hijacking more than 10.000 smartphones with valid SIM modules in order to mount a successful attack.

In this thesis we have described a different approach, we have evaluated the possibility to bypass the strict timings enforced by the cellular network protocols by means of a dedicated radio device. This allowed us to prove that it is possible to inject into the cellular networks signalling traffic at an higher rate than with a standard mobile phone. Given this fact we studied whether unauthenticated devices, that is, devices not controlling valid SIM modules, may reach the same service degradation as a botnet of regular phones: the trade off resides in the fact that while an authenticated mobile station can query the network with high resource-demanding operations, our attacking devices is indeed able to reach higher request rates but, unfortunately, of activities that require less resources on the network side. In this work we have shown that the network carries on expensive calculations, even for unknown device, before actually asking the requesting equipment to commit its own resources. For this reason it is possible to force the radio access interface to inject through the network of a single generation (e.g. 2G, the GSM network) several solicitations, sufficient to produce a significant degradation of the service: this result requires about 1500 dedicated devices that is a reduction of an order of magnitude with respect to the resources employed by attacks described in previous works. In an effort to lower the number of needed devices, we have studied the possibility to hit a single infrastructure core component, the HLR, through different generations of network, thus leveraging the interplay between network generations in the core infrastructure. Our combination of these two factors, using a SIMless dedicated radio device and combining the signalling bandwidth of GSM with the one made available by the 3G (UMTS), allowed us to flood the network with enough requests to clog HLR computing capabilities and also reduce the number of attacking devices from more than 10.000 to barely 400: a reduction in the amount of resources needed for a successful attack that is two order of magnitude lower than the one required in reference literature. Furthermore, our work showed that it is possible to remove even the constraint requiring each attacking device to own a legitimate SIM card. Finally, the device described in this thesis causes a DoS for the signaling capabilities of the cell where it delivers the attack: this last achievement is more effective than using a jammer and uses less devices —that is, one— in respect to previous works.

It is also important to notice that the devices enrolled in a botnet are still positioned by their rightful owners, independently from the attacker will. Thus, it is possible that an unusual clustering of users (e.g. an event in a theatre or a concert) could produce a concentration of devices that saturates the cell signalling bandwidth and prevents some of the botnets node to fulfil their full attacking potential. On the contrary, the device we envision is not owned by an unknowing user, it can be precisely placed by the attacker and even remotely triggered to start the attack. All of these factors represents a significant increase in the dangerousness of the proposed attack when compared with the ones described in previous works.

Finally we want to point out that this thesis trusted exclusively upon measurements and simulations already available in the literature, and, additionally we further elaborated found data to extract some estimations based on theoretical assumptions, although described by standard documentation. Unfortunately there has been no measurement campaign in the wild because, in the first instance, there were no hardware, readily available, that could be used to precisely execute the protocol steps we presented. For this reason the discard of regular phone was beyond doubt because either their protocol implementation is both closed-source and not modifiable or the open source alternatives are limited to the GSM protocol stack.[1] Moreover even if we have had a device meeting our requirements we couldn't have used it because the messages exchange is likely to trigger network alarms that arouse suspicions in the mobile operator, and even will result in fines whenever these practices are illegal. For these reasons we could have reverted to a network simulator but its research resulted only in commercial products because of the span of its elements coverage —from the phone to the HLR— and the need to alter the MS default behaviour.

[1] There is currently the OsmocomBB (`http://bb.osmocom.org`) aimed at developing an open-source protocol stack for GSM.

4.1 Future works

This thesis does not provide a full, in depth analysis of all the problems that the proposed attack arise but some of them are already under research and are left as future works. The main target in the near future will be a detailed analysis of the requirements for the envisioned dedicated device, both in terms of needed hardware and software, which will help also to definitely justify some of the theoretical results drawn by this work.[2] Some of the questions we are going to answer with this in-depth analysis are: "is there any needs of dedicated hardware or is it possible to reuse already available one?"; "from the software point of view have we to develop it from ground up or is there any leakage/open source project we may take advantage of?"; "how much expensive is it?"; "what's about power efficiency?" and so on.

Other interesting results may be found extending the analysis of the air interface also to the newest 4G/LTE network, characterized by a new multiple access technology called Orthogonal Frequency-Division Multiple Access (OFDMA), which may result in ever a lower number of needed attacking devices. Moreover it is indeed useful, mostly from a protection point of view, to characterize the geographical extension of the area affected by the attack, determining the jurisdiction of a single HLR, the total number of cells yonder contained, and thus the ratio of cells affected by the attack.

[2] Obviously also trying not to be put in jail in the charge of terrorism.

Acknowledgements

`\begin{LaTeXtranslate}[enit]`

Questa tesi è il frutto del lavoro di mesi di studio, ricerche e composizione di argomenti e materiali più o meno didattici che coprono *quasi* a 360° il mondo delle reti mobili GSM/UMTS. Mi scuso del *quasi* con i colleghi "telecomunicazionisti" ma certi argomenti "di basso livello" sarebbero stati davvero troppo anche per un informatico prestato alle telecomunicazioni come me. ;) Tutto ciò, comunque, non sarebbe stato possibile senza l'aiuto di tante persone a cui voglio dedicare le prossime righe.

Al prof. Migliardi che, in primis, ha reso possibile questo progetto, in quanto ha saputo stuzzicare fin dal primo colloquio la mia curiosità e poi, con professionalità e competenza, ha continuato su questa strada, indirizzandomi, motivandomi e spingendomi a vedere le cose che scoprivo sotto luci diverse. E poi i prof. Ferrari e Merlo che, fosse anche solo tangenzialmente, hanno partecipato a quanto fatto.

A Matteo Canale e ai professori di telecomunicazioni del DEI, per la loro disponibilità, e soprattutto per avermi ricordato che la vera fonte della conoscenza sono i libri, e non Google (anche se aiuta a trovarli, i libri).

A tutti i colleghi "apprendisti" e "senior" di Telecom per avermi introdotto gradualmente al mondo delle telecomunicazioni tanto da riuscire a farmele apprezzare: sia chiaro, non ho ancora rinnegato l'Informatica, ma in saccoccia ho messo parecchie nozioni nuove e chissà che in futuro possano tornare utili. Tra i colleghi senior un ringraziamento particolare per lo svolgimento di questa tesi va ad Alessandro Pace che ha saputo spiegare in modo semplice e completo concetti che erano sepolti da decisamente troppi strati di polvere.

Ai *butei*, compagni e colleghi di 6 anni di Università, nonché fedeli pranzatori alla Piovego, giocatori di Frozen Bubble, polemici, ma pure pazienti ascoltatori dei miei discorsoni. A Nicola e Claudio, compagni di lavoro, di cene chiacchierose, di annesse serate in Japelli e pure di qualche bevuta. Ad Andrea, che in questi 5 anni di coinquilinismo mi ha sopportato, fatto crescere, e con cui sono riuscito a intavolare discorsi da epopea, nonostante i pochi momenti in cui ci vedevamo prima di andare a letto.

Alla mia famiglia tutta, a cui voglio dedicare questo traguardo, che mi ha finanziato, incentivato e sostenuto in questi sei anni (e forse più) di "devo studiare"; ai nonni che da quaggiù o lassù una preghierina se la sono sempre ricordata.

Infine a Paola che, forse l'unica, è stata davvero partecipe di ogni momento di questo percorso e che, soprattutto, vorrò al mio fianco in tutti i futuri sentieri che la vita ci farà percorrere!

E poi la Qe, le mense Piovego, S. Francesco (di sera, RIP) e Pio, l'aula studio Japelli e tutti coloro con i quali ho condiviso questi anni e che, causa la foga del momento e la stanchezza, non trovano citazione nelle righe precedenti...

Thank you all!

`\end{LaTeXtranslate}`

Bibliography

[1] 3GPP. *TS 23.401 — General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access.* http://www.3gpp.org/ftp/Specs/html-info/23401.htm.

[2] 3GPP. *TS 24.008 — Mobile radio interface Layer 3 specification; Core network protocols; Stage 3.* http://www.3gpp.org/ftp/Specs/html-info/24008.htm.

[3] 3GPP. *TS 25.214 — Physical layer procedures (FDD).* http://www.3gpp.org/ftp/Specs/html-info/25214.htm.

[4] 3GPP. *TS 25.322 — Radio Link Control (RLC) protocol specification.* http://www.3gpp.org/ftp/Specs/html-info/25322.htm.

[5] 3GPP. *TS 27.007 — AT command set for User Equipment (UE).*

[6] 3GPP. *TS 44.006 — Mobile Station - Base Stations System (MS - BSS) interface Data Link (DL) layer specification.* http://www.3gpp.org/ftp/Specs/html-info/44006.htm.

[7] Alessandro Armando, Alessio Merlo, Mauro Migliardi, and Luca Verderame. Would you mind forking this process? A denial of service attack on Android (and some countermeasures). In *Information Security and Privacy Research*, pages 13–24. Springer, 2012.

[8] Aniello Castiglione, Roberto De Prisco, and Alfredo De Santis. Do you trust your phone? In Tommaso Noia and Francesco Buccafurri, editors, *E-Commerce and Web Technologies*, volume 5692 of *Lecture Notes in Computer Science*, pages 50–61. Springer Berlin Heidelberg, 2009. http://dx.doi.org/10.1007/978-3-642-03964-5_6.

[9] K.W. Derr. Nightmares with Mobile Devices are Just around the Corner! In *Portable Information Devices, 2007. PORTABLE07. IEEE International Conference on*, pages 1–5, 2007.

[10] Charalampos Doukas, Thomas Pliakas, and Ilias Maglogiannis. Mobile healthcare information management utilizing cloud computing and android OS. In

Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, pages 1037–1040. IEEE, 2010.

[11] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner. A survey of mobile malware in the wild. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, pages 3–14. ACM, 2011.

[12] Chris Fleizach, Michael Liljenstam, Per Johansson, Geoffrey M Voelker, and Andras Mehes. Can you infect me now?: malware propagation in mobile phone networks. In *Proceedings of the 2007 ACM workshop on Recurring malcode*, pages 61–68. ACM, 2007.

[13] Chuanxiong Guo, Helen J Wang, and Wenwu Zhu. Smart-phone attacks and defenses. In *HotNets III*, 2004.

[14] Gunnar Heine and Matt Horrer. *GSM networks: protocols, terminology, and implementation*. Artech House, Inc., 1999.

[15] Harri Holma and Antti. Toskala. *WCDMA for UMTS*. Wiley Online Library, 2002.

[16] C Johnson, H Holma, and I Sharp. Connection setup delay for packet switched services. 2005.

[17] Georgios Kambourakis, Constantinos Kolias, Stefanos Gritzalis, and Jong Hyuk-Park. Signaling-oriented DoS attacks in UMTS networks. In *Advances in Information Security and Assurance*, pages 280–289. Springer, 2009.

[18] Muzammil Khan, Attiq Ahmed, and Ahmad Raza Cheema. Vulnerabilities of UMTS access domain security architecture. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD'08. Ninth ACIS International Conference on*, pages 350–355. IEEE, 2008.

[19] Nataraj Kuntagod and Chinmoy Mukherjee. Mobile decision support system for outreach health worker. In *e-Health Networking Applications and Services (Healthcom), 2011 13th IEEE International Conference on*, pages 56–59. IEEE, 2011.

[20] Mauro Migliardi and Marco Gaudina. Memory Support through Pervasive and Mobile Systems, in Inter-Cooperative Collective Intelligence: Techniques and Applications. In *Studies in Computational Intelligence*. Springer, 2013.

[21] Collin Mulliner and J-P Seifert. Rise of the iBots: Owning a telco network. In *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on*, pages 71–80. IEEE, 2010.

[22] A. Pace and P. Semenzato. L'Interfaccia Radio UMTS: Approfondimenti su Link Budget, Capacità e Copertura. Dispensa Telecom Italia T5 UMTS.

[23] Marco Petracca, Marco Vari, Francesco Vatalaro, and Graziano Lubello. Performance evaluation of GSM robustness against smart jamming attacks. In *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*, pages 1–6. IEEE, 2012.

[24] Carlo Tacconi, Sabato Mellone, and Lorenzo Chiari. Smartphone-based applications for investigating falls and mobility. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, pages 258–261. IEEE, 2011.

[25] Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Mitigating attacks on open functionality in SMS-capable cellular networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 182–193. ACM, 2006.

[26] Patrick Traynor, Michael Lin, Machigar Ongtang, Vikhyath Rao, Trent Jaeger, Patrick McDaniel, and Thomas La Porta. On cellular botnets: measuring the impact of malicious devices on a cellular network core. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 223–234. ACM, 2009.

[27] Patrick Traynor, Patrick McDaniel, Thomas La Porta, et al. On attack causality in internet-connected cellular networks. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–16. USENIX Association, 2007.

[28] U.S. Department of Defense. *Security Technical implementation Guide*. `http://iase.disa.mil/stigs/net_perimeter/wireless/smartphone.html`.

[29] Mei-Ying Wang, John K Zao, PH Tsai, and JWS Liu. Wedjat: a mobile phone based medicine in-take reminder and monitor. In *Bioinformatics and BioEngineering, 2009. BIBE'09. Ninth IEEE International Conference on*, pages 423–430. IEEE, 2009.

[30] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 46–57. ACM, 2005.