A Denial of Service Attack to UMTS Networks Using SIM-Less Devices

Alessio Merlo, Mauro Migliardi, Nicola Gobbo, Francesco Palmieri, and Aniello Castiglione, *Member, IEEE*

Abstract—One of the fundamental security elements in cellular networks is the authentication procedure performed by means of the Subscriber Identity Module that is required to grant access to network services and hence protect the network from unauthorized usage. Nonetheless, in this work we present a new kind of denial of service attack based on properly crafted SIM-less devices that, without any kind of authentication and by exploiting some specific features and performance bottlenecks of the UMTS network attachment process, are potentially capable of introducing significant service degradation up to disrupting large sections of the cellular network coverage. The knowledge of this attack can be exploited by several applications both in security and in network equipment manufacturing sectors.

Index Terms—Mobile security, UMTS, cellular networks security, HLR, DoS attack, critical infrastructures

1 Introduction

Mobile phones based on cellular networks are one of the most successfully deployed technology of the last decades and coverage of cellular networks in the world has generally become pervasive. Both an effect and a cause of this success may be seen in the evolutional cycle of the network technologies.

In fact, while the evolution from early analog networks to recent 3G/4G solutions has allowed Mobile Network Operators (MNOs) to offer new services to their customers, the same time it has pushed new needs into the customers that, closing the cycle, require more resources to be supported. As an example, we may observe how the user needs have evolved from simple voice and short text message communications to high speed Internet connections and ubiquitous access to multimedia streams and storage repositories made possible by the introduction of General Packet Radio Service (GPRS) allowing data delivery according to both the circuit and packet switched paradigms.

In this scenario, mobile communication networks have gained the role of critical infrastructure for the global community like transportation or electricity so that many individuals and business activities relying on them for their day-to-day operations may be severely impacted by any service degradation or disruption. It is thus critical to tackle the problem of security in mobile networks from every possible perspective, not only focusing on the confidentiality and integrity of codes [1], end-to-end connections [2], [3], information flows [4] but also considering the availability of the network itself.

The complexity of the mobile network structure may hide both unknown and known vulnerabilities that proper analysis tools and formal techniques (e.g., [5]) can unveil.

Beyond protocol-specific vulnerabilities, the same network complexity may also hide potential performance bottlenecks in signaling protocols or control applications/components that can be exploited by several kinds of Denial of Service (DoS) attacks in order to tear down critical service subsystems or overwhelm them with large number of requests, exhausting the resources needed to ensure network operations.

The effects, in terms of coverage, of DoS attacks progressively increase when moving from physical (i.e., using a radio jammer) towards the upper layers (i.e., affecting application-level subsystems serving large portion of the cellular network).

Luckily, most of the known attacks are not easy to implement since they require a very large number of mobile cooperating devices (usually several thousands) or access to internal MNO facilities to be really effective. Nonetheless, the potential impact of these attacks on mobile phone networks has not been sufficiently assessed and needs further study.

To this aim, this work, by focusing on the *node attachment procedure* in Universal Mobile Telecommunications System (UMTS) infrastructures, shows that it is possible to mount a full-fledged DoS attack potentially capable of shutting down large sections of the network coverage without the need of hijacking or controlling actual users' terminals, as well as that the number of devices necessary to make such an attack effective is limited to a few hundred ones.

This attack exclusively operates at the user-level by relying on unavoidable protocol-level signaling features so that no hacking on intra-operator facilities is needed. It is indirectly targeted at the Home Location Register (HLR)

- *A. Merlo is with the University of Genova, Italy, and the E-Campus University, Italy. E-mail: alessio.merlo@unige.it, alessio.merlo@uniecampus.it.*
- *M. Migliardi and N. Gobbo are with the Department of Information Engineering at the University of Padua, Italy. E-mails: {mauro.migliardi, gobbonic}@dei.unipd.it.*
- *F. Palmieri is with the Department of Industrial and Information Engineering, Second University of Naples, Italy. E-mail: francesco.palmieri@unina.it.*
- *A. Castiglione is with the Department of Computer Science, University of Salerno, Italy. E-mail: castiglione@ieee.org, castiglione@acm.org.*

Manuscript received 14 Jan. 2014; revised 23 Mar. 2014; accepted 26 Mar. 2014; date of publication 1 Apr. 2014; date of current version 14 May 2014. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TDSC.2014.2315198

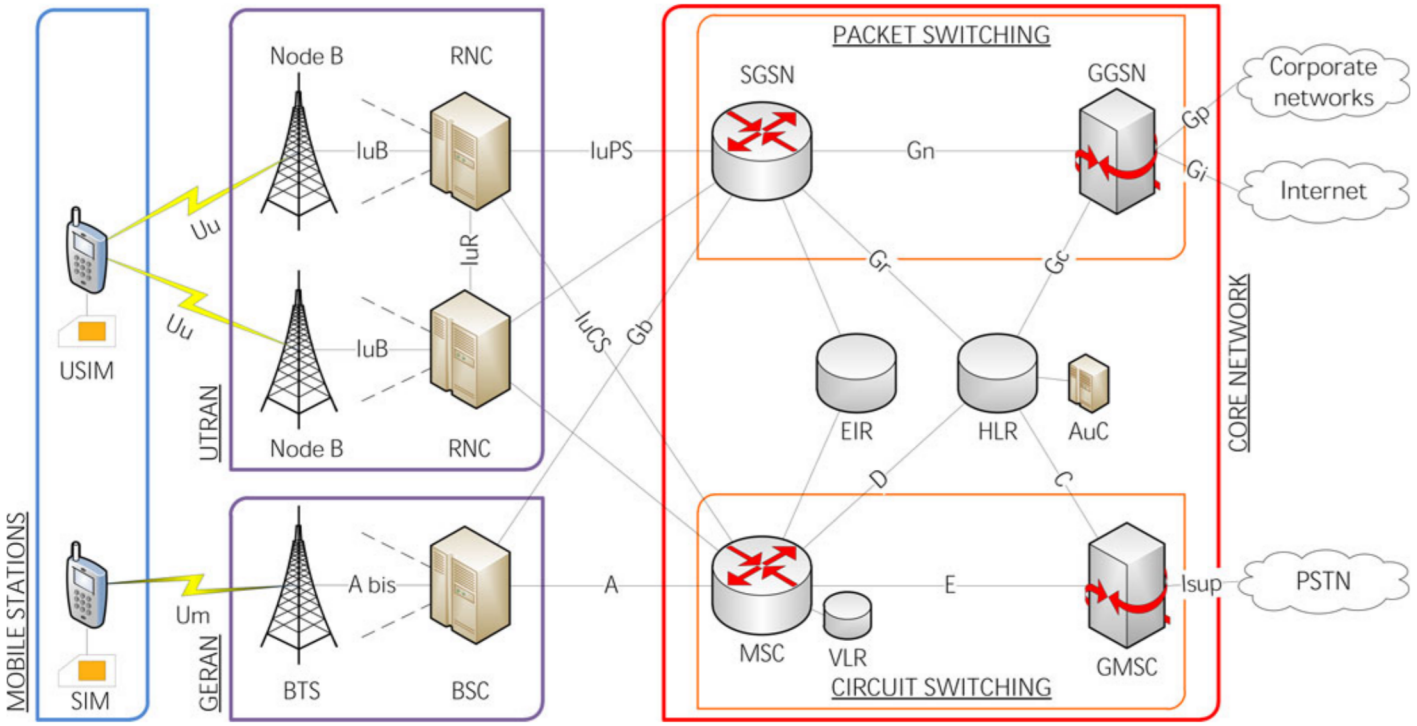

Fig. 1. UMTS standard network representation.

that is the database containing information on mobile subscribers as well as call blocking and forwarding rules, that can be overwhelmed by service requests.

Since this database is a critical component, often revealing to be a major bottleneck within the overall infrastructure, an outage of its functionality may cause an interruption of other mobile services too, finally resulting in a mobile network DoS potentially leaving thousands of devices without their lifelines to the network core.

Furthermore, the presented attack does not require the use of real mobile handsets equipped with valid Subscriber Identity Module (SIM) modules and needs only a limited number (a few hundreds) of UMTS radio interfaces, eventually located on a single ad-hoc device, in order to inject the signaling traffic necessary to reach a critical level of disruption on the target cellular infrastructure.

Such activity, being not constrained by the fixed operation timings characterizing standard SIM-empowered COTS devices, may also easily disable the signaling capabilities of the cells under attack, causing a local DoS similar to the one that can be achieved with a radio jammer.

In order to correctly dimension the resources needed to make the aforementioned attack really effective, the amount of signaling traffic that a simpler and faster dedicated SIM-less device can generate (by pushing UMTS interfaces to their design limits, as well as its effects on the critical network core components) have been analyzed by leveraging the HLR performance measurement available from [6].

The ideas presented in this work can give rise to several applications, ranging from cyber-warfare devices, that can be used in both intelligence and military scenarios to temporarily defeat UMTS communications within specific areas, to assessment/benchmarking tools that can be extremely useful in dimensioning, through "torture test" practices, new distributed HLR solutions.

2 Introducing the UMTS Network

A typical UMTS Public Land Mobile Network (PLMN) architecture (see Fig. 1) is divided into three main building blocks: mobile station (MS), UMTS Terrestrial Radio Access Network (UTRAN) and core network (CN).

The MS or user equipment (UE) may be a mobile phone/terminal or a mobile broadband modem providing UMTS protocol stack and radio access capabilities. It is marked with a worldwide unique identifier, called International Mobile Equipment Identity (IMEI) and equipped with a SIM in order to allow end user identification and authentication based on a unique subscriber identifier, the International Mobile Subscriber Identity (IMSI), together with its associated private cryptographic key. At the network attachment time, the IMEI is checked against the equipment identity register (EIR), in order to banish stolen or out-of-requisites hardware from the network. Furthermore, being the IMSI univocally associated to a subscriber, in order to avoid its use as a way to track users in their movements by unlawfully eavesdropping radio traffic, another identifier called Temporary Mobile Subscriber Identity (TMSI) is used. In detail, once the device has been switched on and during the preliminary messages exchange, a new TMSI value is calculated. The TMSI value has just a local validity, it is often refreshed with a new one and it is used in each communication from and towards the network. The use of fresh and ever-changing TMSI values allows to get a high degree of anonymity and robustness against eavesdroppers.

The UTRAN, organized in grouped cell towers known as Node B stations, is equipped with one or more antennas, is connected to a radio network controller (RNC) component and is responsible for radio resource and mobility management as well as encryption of user's data. The RNC improves the base station controller (BSC) functions provided in the traditional GSM/EDGE Radio Access Network

(GREAN) (manly interconnecting the different base transceiver station (BTS)), by also supporting the IuR interface that allows RNC-to-RNC communications. This UMTS novelty, along with its specific protocol peculiarities, permits soft handover, that is, a feature where a MS can be simultaneously connected to two or more cells, in order to maximize the received signal quality.

The CN connects each RNC to the Serving GPRS Support Node (SGSN) and to the mobile switching center (MSC), in order to transport, respectively, packet and circuit switched information. MSC and SGSN also interconnect the UTRAN with the traditional GREAN acting as both switching and termination point for end-to-end connections between the different domains by managing handovers between BSC/RNC nodes. These components are also involved into the authentication procedure and the mobility/location management by continuously keeping track of MSs movements within their coverage areas. Such operation requires the presence of an auxiliary database known as visitor location register (VLR) containing the user identities associated to their current RNC-level location, together with a pointer to the MNO's main subscriber record which is kept into the HLR, behaving as an anchor point to which the MS remains stably tethered while moving from cell to cell. In detail, the HLR maintains a record for each mobile subscriber including several details such as the telephone number, IMSI and related cryptographic information, call blocking and forwarding rules and a pointer to the most updated VLR the user is known to be roaming on. It relies on the Authentication Center (AuC) for calculating the challenges and responses needed during mobile user validation.

From the above description, it is easy to see that HLR is the most complex and strategic network component, usually serving large portions of the mobile infrastructure (involving thousands or millions of mobile users), that has to be always queried for both phone call and data/SMS delivery, as well as for billing and authentication procedures.

3 Related Work

Both the US and the European institutions include cellular networks in the vision of critical infrastructure for Homeland Security [7], [8]. This new level of attention has spurred a large number of research groups in performing deep and accurate studies on the security of such infrastructures from the availability point of view. Furthermore, new tools for threats management have been devised [9]. Many of these studies have led to the discovery of sweet spots for attacks and vulnerabilities that could be exploited to threaten the actual network availability. At the same time, though, the network architecture and its components have been deeply analyzed and solutions to eliminate, or at least mitigate, the exploitability of these sweet spots have been devised.

Two main factors introduce an additional level of complexity in the vulnerability of the cellular infrastructure as a whole and thus in the measures needed by its defense, namely i) the complexity and the high level of programmability of current mobile phones (i.e., smartphones) and ii) the interconnection between the cellular network and the Internet. Regarding the former aspect, actual smartphones are complex devices that may both suffer from intrinsics vulnerabilities [10] and allow average programmers to easily build malware [11]. More specifically, it has been widely recognized that the set of open features provided by modern smartphones makes them the most suitable choice for massive and distributed mobile network attacks [12]. In fact, a malicious smartphone 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 (BS), can jeopardize its availability by making no-answer calls and thus saturating provisioned user-plane voice channels. The same authors also demonstrate that if the subverted smartphones are not in the same place, it is still possible to stop the voice services by performing a distributed DoS, but exploiting a higher number of compromised devices.

On the other hand, the interconnection between the mobile network and the Internet increases the possibility to perform attacks to the cellular network [14], [15]. In case of a DoS attack, such interconnection may improve the chance to perform successful distributed DoS attacks to the cellular network by means of *botnets* of compromised mobile devices. However, mobile networks have constraints and characteristics that should be taken into account by attackers mounting a botnet, both during the infection phase [16] and the setup of the command-and-control mechanism [17]. As reported in [18], an attacker able to control a botnet can use infected devices for multiple purposes like spam delivery, sending SMS to premium services, remote wire-tapping, just to cite a few.

Later studies on DoS attacks have shown that it is possible to achieve a high level of service degradation in a more efficient way than consuming voice and SMS traffic (i.e., user-plane channels). In fact, an attacker may try to flood control channels, which are separated from traffic ones and very limited in terms of available bandwidth. The first work in this direction [19] shows how the interconnection between mobile networks and the Internet can be exploited by an attacker continuously sending SMS from an online service to a crafted hit-list of telephone numbers; the generated data flow is sufficient to keep control channels, shared by SMS and voice, completely busy.

The works presented in [20] and [21] respectively examine some UMTS protocols flaws that can be used to delete, modify or replay some unauthenticated or not integrity-protected messages in order to launch DoS attacks against both user phones and network nodes. These flaws may also permit revealing user identities (IMSI) and impersonating the network performing man-in-the-middle attacks. Also, as described in [20] the way a DoS attack to the UMTS network can be mounted is straightforward if the attacker disposes of a list of valid IMSI. In this case, it is sufficient to send *attach requests* with a valid IMSI to force the UMTS number to start the protocol for attaching the device. Previous studies, however, have two limitations. First, in many cases (e.g., [20], [21]) authors do not evaluate the amount of resource needed to actually mount the proposed DoS attacks as, instead, it is performed in detail in [6] and [22] for the GSM network. Moreover, analytic assessments indicate that, in order to perform effetive DoS attacks at the UMTS network, thousands of compromised devices may be required [6].

In the following we show how it is possible to leverage the above ideas as well as the characteristics of the UMTS standard to greatly reduce the amount of resources needed for a successful DoS attack against UMTS infrastructures.

4 Attacking the UMTS Network

Telecommunication companies envision a mobile ecosystem biased toward network-centric intelligence, without taking advantage of today's smartphones capabilities.

In fact, even if the new generation mobile terminals are more intelligent and powerful than their predecessors, networks infrastructures still do not make use of these enhanced features by assuming the lowest possible capabilities in order to ensure backward compatibility with older devices. This implies that access procedures are made computationally light for the terminals by delegating to the network most of the operations and resources requirements. This, as a consequence, results in higher signaling traffic levels between network nodes and in far more complex signaling protocols and management issues.

These facts are the basis of the vulnerabilities that can be exploited to introduce infrastructure-level DoS by tampering the network attachment process.

4.1 Exploiting Radio Access Protocols

When a mobile cellular device is powered on, the UMTS protocol defines the actions that should be carried on in order to *attach* to the network.

A high level description of the network access procedure can be sketched in the following common steps: i) cell discovery, ii) best server synchronization, iii) attachment request, iv) authentication and key agreement (AKA) and v) temporary identity creation. The peculiarity of this procedure is that it cannot leverage previously accrued knowledge since it must accommodate for new devices about which there is no previous information. In such a context, the most critical step from the DoS vulnerability perspective is the AKA one, where an unauthenticated device may force the core network to carry on computations that are more resource-consuming than the connection request itself.

As described in [20], the way an attack based on the above considerations could be set up is straightforward: in a preliminary phase an attacker builds a database of valid IMSIs, then, he floods the network with multiple *attach requests*, each one carrying a different IMSI chosen from that database. The cellular network, in order to accomplish the AKA phase, forwards the requests to HLR/AuC where each IMSI is validated and, being authentic, triggers the calculation of authentication information that are sent to SGSN that, in turn, has to submit the challenge back to the MS and verify the correctness of the reply. As the attacker does not really own the SIM corresponding to the used IMSI, he does not know the correct answer. However, such a correct answer is not strictly needed; in fact, since the attack's goal is to exhaust HLR/AuC computing resources, hitting the target with a sufficiently large volume of valid *attach requests* is enough to introduce network core level DoS.

4.2 Estimating the Effects on Core Components

The work [20], however, does not provide an assessment for the HLR/AuC performance impact, thus they do not estimate the number of terminals needed by an attacker in order to considerably degrade HLR services, that is the most strategic component affected, by using the attack described above.

A partial analysis of this problem is provided in [6]. In this work Traynor et al. 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, authors measure the average throughput—in transactions per second (TPS)—of a HLR setup, with respect to different transaction types. They choose the *insert call forwarding* procedure as the attack vector because it offers the best tradeoff between computational load and execution speed.

A standard mobile phone controlled via the AT interface has been used to simulate the effect of injecting attack traffic on a HLR already serving a typical mix of transactions. In this way, they were able to find that when injecting 2,500 TPS with a request period of about 4.7 s, the HLR capability of managing legitimate requests—under low-traffic assumptions—is reduced by 93 percent. In particular, the above request period is composed by 2.7 s spent in executing the insert call forwarding transaction, whether the remaining 2 s are a delay guard between successive requests, needed by the device to operate correctly.

From the results presented in [6] it is possible to determine the rate of malicious requests R (TPS) that an attacker is supposed to deliver in order to achieve a target HLR throughput degradation. Once this rate is defined, the following equation gives the number of compromised devices N needed to make the attack really effective, by using a request period τ:

$$N = R \times \tau. \tag{1}$$

It is also possible to infer that the *get access data* procedure is roughly five times faster than the *insert call forwarding* one. So, in order to achieve the same level of service degradation, we assume that the attack traffic must be multiplied by 5. This puts our target to 12,500 TPS.

However, this is a worst case scenario for the attacker: in fact, the tests presented in [6] focus only on the HLR, disregarding the computations at the AuC that is needed to determine the authentication information.

Since, for obvious security and law enforcement reasons, attacking a real "production" HLR is not feasible and obtaining a test HLR from MNOs to perform attack performance measurements is very difficult (if not impossible), the results presented in [6] remain the only reference data available for studying and estimating the effects of the aforementioned attacks on HLRs. However, while several advancements in architectural design took place in the last years for many components within the network core, HLR architecture has remained almost unaffected for decades, even if the most recent HLR systems scaled several orders of magnitude over their predecessors in order to support the volumes characterizing modern networks.

The last generation of HLRs is implemented on clustered systems with multiple front-end units to process network-

level signaling and multiple replicated databases in order to reliably store the subscriber information. Cluster components are geographically distributed across multiple sites on the operator's network.

This introduces improved resilience features at the expense of added complexity so that if the entire distributed system fails it can take longer time to be back in service. Furthermore, a much higher amount of users can be handled by a single system so that an outage, while less probable, can potentially affect a larger section of the cellular network.

In addition to the large amount of computing power available in modern clustered HLRs, it is also necessary to consider the counter-intuitive result outlined in [6] showing that the more busy the HLR is, the more difficult is disrupting its services.

The explanation 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 higher their probability of being processed is, that is, only a more powerful attack may convey enough malicious requests so they are more likely to be served instead of legitimate ones.

4.3 Complexity Issues in Hybrid Mobile Networks

Another significant security facet of cellular network is the fact that the deployment of new faster and smarter technologies cannot discontinue the support of the older ones since also in presence of a significant technological improvement, a certain number of legacy devices remains active for a long time, due to physiological inertial phenomena according to which many users avoid substituting their devices until they keep working and some manufactures keep producing legacy terminals for low-end market needs.

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.

To cope with this need, network operators adopt single Radio Access Network solutions, where a cellular site broadcasts signals related to up to three different technologies in five different frequency bands.

Besides this composite network architecture, there is also the tendency of telecommunication companies to concentrate the intelligence of the system in the network itself disregarding the growing capabilities of the new generations of mobile terminals.

While it makes easier to maintain the network compatible with older devices, this assumption results in higher signaling traffic levels between network nodes[1] and far more complex (and vulnerable) signaling protocols and management issues. This makes the network-level facilities the ideal targets for DoS attacks.

4.4 Why Regular Devices May Become a Limiting Factor

Botnets of mobile nodes are the most useful tools providing suitable features for originating DoS attacks. To launch the attack presented in [6] a sufficiently large smartphone botnet is needed for two reasons: first, clients must be authenticated before submitting an *insert call forwarding* request; second, this 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. 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. Of course, the SIM must be associated with a valid subscription, otherwise, in presence of an invalid SIM the device is able to initiate the attach procedure, but the network rejects it without needing a significant amount of resources.

However, also in case of a valid SIM, allowing completion of the attach procedure, the solution is definitely suboptimal because of intrinsic features of the phone device itself. The built-in mobile protocol stack is implemented strictly following 3GPP specifications which, in turn, are full of transmission wait timers, exponential backoffs, maximum re-transmission trials and other artifices [23] designed with the precise purpose of implicitly inducing a fair use of the network resources.

As a proof of this fact, the experiences presented in [6] highlighted that, during network behavior measurements, a 2 s delay was necessary between each request: its removal, otherwise, caused extended execution times.

The real goal of a DoS attack, on the contrary, is to unfairly squander the network resources in order to prevent legitimate devices from accessing the service. Furthermore, we want to reach the limits of the air interface in order to cut down the number of attacking points.

For these reasons we claim that the tool best suited to an attacker needs is a dedicated device capable of accessing the network without requiring a valid SIM, and without the timing guards and strict protocol adherence that are normally introduced in components aimed at the consumer market.

5 Assessing the UMTS Radio Interface

We now analyze the peculiarities of UMTS radio interface at the protocol layer in order to evaluate its potential attack surface and limits in terms of number of *attach requests* sent to a Node B station per second.

In this process we suppose to be the only device communicating with the target cell. This hypothesis seems unrealistic, but is a direct consequence of the unfairness of the attacking device: while legitimate mobile phones would backoff when facing a traffic problem, the attacking 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 will not 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 are likely to allow the attacker to grab the resources just freed, making it unavailable to legitimate devices.

1. http://connectedplanetonline.com/mss/4g-world/the-lte-signaling-challenge-0919/ (accessed in January 2014).

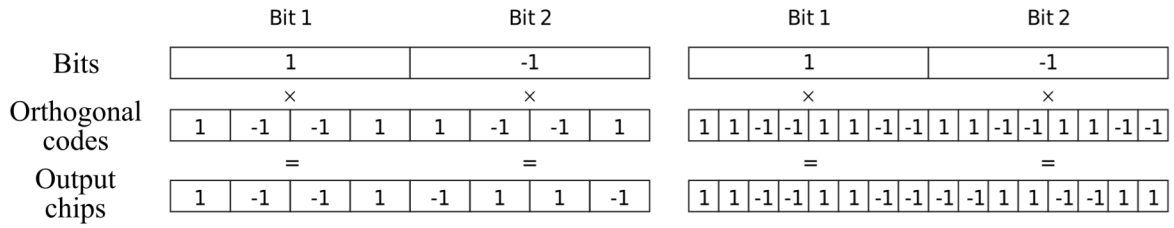

Fig. 2. Different *spreading* outcomes obtained by multiplying the source signal with Walsh-Hadamard sequences.

5.1 Channel Coding in UMTS

The UMTS mobile cellular system relies on Wideband Code Division Multiple Access (W-CDMA), allowing Node B to transmit simultaneously to multiple mobile phones on the same carrier frequency as long as different *channelization codes* are used.

These codes—also known as Walsh-Hadamard sequences—are multiplied by the bit sequence coming out from the channel coding block: the resulting sequence has a higher rate than the input one and UMTS specification fixes it at 3.84 Mcps—where the "c" stands for *chip*. In Direct-Sequence CDMA, the user signal is multiplied by a pseudonoise code sequence of high bandwidth. This code sequence is also called the chip sequence and a chip is the sub-period in which the pseudonoise code sequence cannot change. Thus while the original signal does not change for the bit period, the coded signal does not change only for the shorter chip period. The resulting coded signal is transmitted over the radio channel. 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 Fig. 2, 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.

However, the most important property belonging to Walsh-Hadamard codes is *orthogonality*, meaning 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 value equal to zero.

In order to obtain orthogonal codes with different lengths, the "binary tree-rule" method is used. Such method is depicted in Fig. 3 and described by the following recursive equation:

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

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, according to the UMTS documentation, the code lengths are in the range 4-512, thus further reducing the choice.

However, not all codes 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 as 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 fact opens the problem, on the Node B side, to distinguish among different MSs. This task is accomplished by *scrambling* codes. This also allows distinguishing between 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 by the signal after spreading codes but, being 38,400 chips long with a repetition of 10 ms, the resulting time rate is not changed. Their generation process uses a pseudorandom number generator which makes this codes *uncorrelated*.

This characteristic is looser than orthogonality, resulting in a much higher number of available codes, but also causes a certain amount of interference between signals since the multiplication of two scrambling codes bit-by-bit and summing up the obtained results gives a total that is zero only on the average. This leads to a higher chance of getting errors as new devices join the network.

5.2 Limits of the UMTS Attack

Focusing on the message exchanges between MS and Node B during the attack, as illustrated in Fig. 4, the network

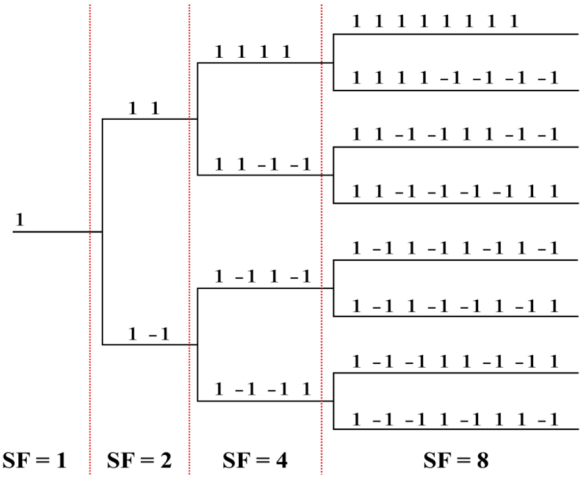

Fig. 3. Portion of the spreading codes tree: UMTS uses lengths in the range 4-512.

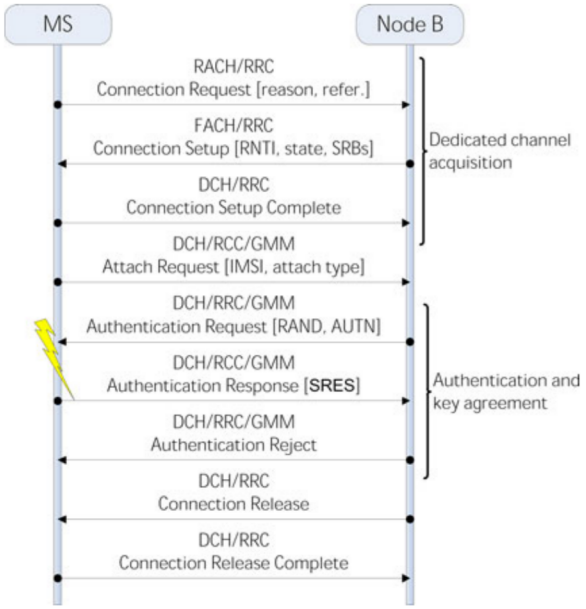

Fig. 4. Messages exchanged during the attack.

attachment, or *location update* procedure, mainly involves three channels. The RACH is the uplink used to carry mobile device's access requests.

The FACH is used to answer incoming random access requests; it carries the information needed by the mobile phone to access the dedicated channel (DCH) used for further communications.

The main part of the procedure is delivered via a DCH bidirectional channel assigned to a mobile terminal, and is reserved to it until a special *release* message is issued by the Node B. In order to evaluate the design limits of the UMTS protocol, we need to analyze each channel and to find out which one introduces the narrowest bottleneck.

In order to enhance the effectiveness and efficiency of our attack, we follow the standard UMTS *location update* message flow only for the first part of the procedure. The first message that deviates from a standard UMTS *location update* message flow is the authentication response message. In this case 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 [24], nor at RLC one [25].

The attack vector of Fig. 4 is exactly the same described in [20] and it is the one using as few Node B/SGSN resources as possible in order to avoid making the processing power of these devices an unintentional bottleneck.

This approach is the exact opposite of the one used in the attack described in [21] that aims at introducing excessive stress both in HLR and in SGSN: the authors modify the MS capabilities declared in the initial *GPRS attach* message; in this way, the *location update* procedure execute flawlessly until the *security mode* command is issued, that is, when the MS checks the security capabilities previously received by the network and, recognizing the inconsistency, terminates the procedure.

We argue, however, that trying to obtain also a SGSN DoS requires defining the attack very carefully because, otherwise, the same device may easily become an obstruction for requests targeted at the more capable HLR. The work in [21] also proposes to couple its attack with a database of stolen IMSIs in order to cause a much more serious damage: this is however impossible because the attacker without knowing K_i would not pass the authentication phase and hence, would not reach the *security mode* command needed to trigger the additional resource depletion.

Before further discussing the attack, a precise estimation of how long dedicated resources are kept occupied by a single request is necessary. To this end, the study reported in [26] profiles 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.

The above analysis covers both a SRB capability of 3.7 and 14.8 kbps. An UMTS Radio Bearer is a data streams that spans multiple network elements with a defined quality of service (QoS), bit-rate, acknowledgement mode and other parameters defined both by the official documentation and network planners.

Radio Bearers reserved for signaling are typically declined with the two bit-rates stated above: the 3.7 kbps is the most common one because it requires less resources, but, when signaling traffic gets higher, switching to the more capable and more resource-expensive 14.8 kbps SRB may become necessary. The timing measurements presented in [26] show that the MS receives the *security mode* command at $1{,}160$ and 850 ms respectively: this message is what a mobile device usually receives after it passes the authentication phase.

In our scenario it will be replaced by the *authentication reject* dispatch, followed 10 ms later by the *rrcConnectionRelease*: this 10 ms delay is due to the transmission time interval (TTI) of the UMTS signaling frame, assuming a channel without jitters. 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.

Finally, 10 ms after the connection release request, the MS replies with a *rrcConnectionReleaseComplete* roughly at $1{,}180$ ms for the 3.7 kbps case and 870 ms for the other. These values, however, do not include the HLR/ AuC interrogation overhead that authors estimate being 600 ms, thus resulting in a total procedure time of $1{,}780$ and $1{,}470$ ms for the 3.7 and 14.8 kbps, respectively.

It is important to notice, however, that these timings may be overestimated in our scenario, because the *security mode* command forces network elements to activate ciphering and integrity protection routines: this overhead is obviously not present when the authentication request is rejected. The high-level description of UMTS already defines two of the three constraints that limit the number of users a Node B may concurrently manage: channelization codes and interference. The third constraint that remains to be mentioned is the network access.

We will now analyze these three aspects to identify the most stringent one in terms of attacking capacity.

5.3 Random Access Analysis

The first UMTS bottleneck we take into account is RACH.

Before accessing the RACH, the MS has to send out some short preambles, with increasing power, until Node B acknowledges the reception over the Acquisition Indicator Channel (AICH): the procedure is defined in this way in order to select the minimum power needed to reach the Node B itself.

Preambles consist of 256 repetitions of a 16 chips long Walsh-Hadamard sequence: in this way, the MS may randomly choose among 16 sequences. Once the output power has been calibrated, the mobile phone may transmit its single transport block message over the RACH, which usually takes a 20 ms transmission time interval.

Sticking to the single-device hypothesis, and considering that, being the attacking device stationary, it takes just one preamble to obtain Node B attention, we estimate a total RACH utilization time of 30 ms. This estimation, however, is based also on the assumption that the *rrcConnectionSetupComplete* message is not sent over RACH because MS early declares its attach intentions in the *rrcConncectionRequest*. In this way, the network is likely to redirect the high amount of successive signaling traffic over the DCH, instead of keep polluting the shared one.

This analysis, coupled with the AICH capability to acknowledge up to 16 preamble signature at the same time, leads to a random access capacity of:

$$\rho_{RACH} = \frac{16}{30 \text{ ms}} \approx 533 \text{ TPS}. \tag{3}$$

5.4 Forward Access Channel Analysis

Once the network has received the *rrcConnectionRequest*, it assigns dedicated resources via *rrcConnectionSetup* message sent over the FACH, a shared downlink channel. This message is relatively large as it typically requires seven transport blocks of 168 bits each, transmitted by multiplexing them in couples, using 10 ms TTI [26]. This leads to a total capacity of the FACH channel of:

$$\rho_{FACH} = \frac{1}{7/2 \times 10 \text{ ms}} \approx 28.5 \text{ TPS}, \tag{4}$$

which is also consistent with the FACH throughput of 32-33 kbps commonly provided in literature.

5.5 Downlink Network Segment Analysis

When the 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 users is limited only by the cell transmitting power and by the number of available channelization codes.

The transmitting power is not a major constraint considering that the attacking device will be placed near the antenna and it does not move.

On the other hand, channelization codes are a scarce resource and, in order to estimate the number of available ones, we have to formulate an hypothesis about the SF (number of chips sent for each bit of information) used by dedicated channels. Given that uplink throughput is usually lower than downlink one, we use user-layer uplink DCH data rates calculated in [27] 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 $\text{SF} = 128$ [27].

TABLE 1
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
$\rho_{\text{DLchannel}}$ (Transactions Per Second (TPS))	132.6	80.3

We are now able to derive the downlink channel capacity ($\rho_{\text{DLchannel}}$) by 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 whenever the MS receives the *rrcConnectionSetup* message, its dedicated channel has been already reserved. Results taking into account all these factors are presented in Table 1.

5.6 Uplink Network Segment Analysis

The uplink segment uses scrambling codes to distinguish between transmissions coming from different MSs. These codes, however, cause interference with each other, thus it is not possible to arbitrarily add new MSs, trying to exhaust all available scrambling codes. For this reason, CDMA networks are referred as *interference-limited* systems.

The estimation of the number of devices 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-orthogonality, approximated by scrambling codes, it can be written as

$$\text{Pole Capacity} = \frac{W}{R_b} \times \left(\frac{E_b}{N_0}\right)^{-1}, \tag{5}$$

where W is the chip rate fixed, in W-CDMA, to 3.84 Mcps, 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, transmission characteristics like receiver sensitivity, channel description, modulation and channel coding types have to be taken into account.

For our calculation, we considered an $E_b/N_0 = 6$ dB which is 1.5 dB higher than the state of the art estimation for a voice uplink "pedestrian" channel presented in [27].

Pole capacity, however, is just a theoretical limit because the uplink noise rises as $(1-\eta)^{-1}$, with η indicating the cell load factor. This means that when η approaches 1, also the power needed to keep the same E_b/N_0 at receiver side moves toward infinity. This phenomenon is called *ROT* and forces the system to work away from its analytical limit: typical configurations account for a maximum load factor of $\eta = 75\%$ [27]. By composing the above constraints the uplink channel capacity ($\rho_{\text{ULchannel}}$) can be estimated, as presented in Table 2.

TABLE 2
Uplink Attacking Capacity Calculations

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

In estimating the uplink capacity calculations, we have ensured that ROT capacity does not exceed the number of available downlink dedicated channels; indeed, the comparison between Tables 1 and 2 confirms our results.

Another interesting side note to uplink and downlink calculations concerns the higher attacking capacity of the 3.7 kbps SRB with respect to the 14.8 kbps one.

The 3.7 kbps SRB is indeed slower in performing *location update* signaling than the other one but Tables 1 and 2 show that the latter has a lower "attacking efficiency" because it requires additional resources not justified by 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. This result, however, indicates that it is possible to mount an attack with just 446 devices instead of more than 11K as described in [6].

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.

Our attack requests, on the other hand, are fast and bandwidth-limited, that is, right in the opposite direction from typical UMTS transactions and, as a direct consequence, they clash with the increased connection setup complexity.

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

6 Doubling the Attack Power Using SIMs

To overcome the limitation described above, we further investigated the UMTS specifications covering the attach procedure [24], looking for any stratagem that would force the core network to query the HLR more than once before tearing down the ongoing signaling connection.

The security of UMTS has been improved under many aspects with respect to the previous generation network, and some of these (e.g., network authenticity checks) are completely new.

Testing the authenticity of the network allows a MS to discover 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. This value is sent with the *authentication request* message and obtained as described in Fig. 5.

This generation is based on a pseudorandom value RAND, on the authentication and key management field AMF that contains some information regarding the MS network validation tolerance and key lifetime, as well as on the IMSI

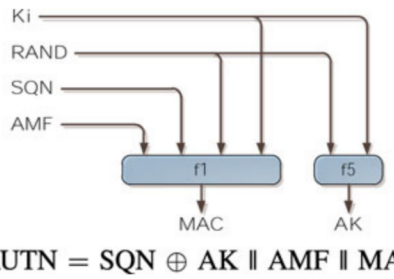

Fig. 5. Information involved in calculating AUTN value.

secret key $\texttt{K}_i$ and a sequence value SQN which is incremented after each successful authentication. These last two information are kept strictly secret by MNOs so that only a legitimate network that knows both of them could create a valid AUTN.

The MS may incur in different failures during the AUTN check; one of them is related to the SQN value being out of the correct range, which in turn leads the MS to inform the network about detected problems 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 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 an *authentication request* with one of the freshly obtained authentication vectors to the MS.

This process, however, may be executed only once because 3GPP documentation [24] explicitly states that the network may terminate the authentication procedure if two consecutive *authentication failure* messages are received. The way in which an attacker may take advantage of this is straightforward as reported by the message exchange depicted in Fig. 6.

Despite the attack simplicity, the message exchange specifies that the *authentication failure* message carries also the AUTS value along with the justification.

The AUTS value 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 due to the requirement of a valid SIM card.

Fig. 7 explains how AUTS is calculated and shows that, as long as requisites of functions $f1$, $f1^*$, $f5$ and $f5^*$ hold, it is robust against the 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 the $\texttt{SQN}_{\text{MS}}$—is concealed using both $\texttt{K}_i$ and RAND;
- *tampering*: AUTS is authenticated with IMSI private key.

The attack capabilities characterizing 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 in the standard attack, we state that current FACH capacity doubles the old one.

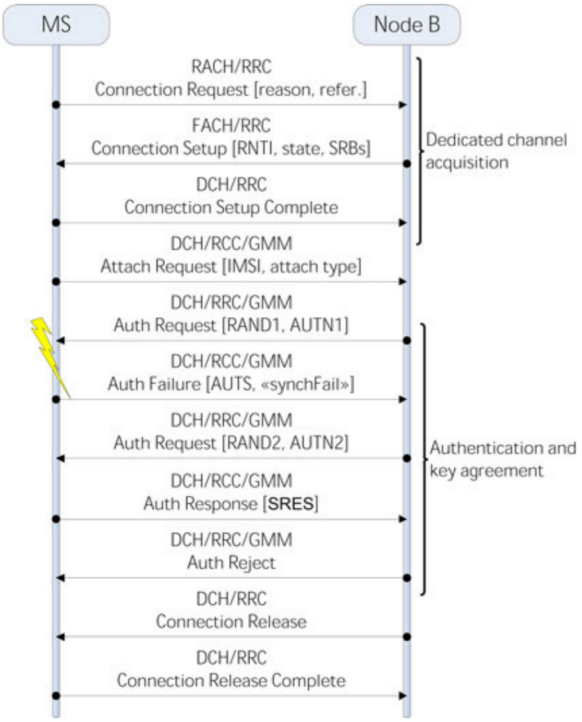

Fig. 6. Messages exchanged during an attack using the *synchronization failure* strategy.

The reason is that for each RRC connection set up, the attacker is able to query the HLR twice, resulting in this channel carrying up to

$$\rho'_{FACH} = \frac{2}{7/2 \times 10 \text{ ms}} \approx 57.1 \text{ TPS}. \quad (6)$$

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 the two message exchanges of Figs. 4 and 6 we note that, with respect to the standard attack, the "synch failure" just requires another full authentication phase plus the *authentication failure* message.

We have shown that the former takes about 600 ms, while for the latter we estimate about 100 ms, which represents a conservative average of the message delivery timings profiled by [26]. This assumption leads to a total execution time for the "synch failure" attack of 2.48 s.[2] Then, focusing on the uplink network segment, which is the most constrained one, its capacity becomes:

$$\rho'_{\text{ULchannel}} = 2 \times \frac{195}{2.48 \text{ s}} \approx 157.2 \text{ TPS}, \quad (7)$$

that still represents an improvement over the 109 TPS of the 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 percent, therefore resulting in higher efficiency also for dedicated channels.

2. We refer to the 3.7 kbps SRB case only, because we already pointed out how the 14.8 kbps SRB proved to be less efficient.

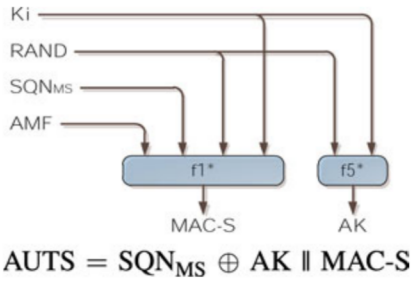

Fig. 7. Information involved in calculating AUTS value.

This analysis shows that, in order to grasp the full potential of the *location update* procedure, the attacker may use SIMs, therefore doubling the number of requests sent per second, resulting in only one half of the devices required to mount the attack in the SIM-less case, namely 223 devices.

This result shows that it is possible to produce a DoS attack to the cellular network with two order of magnitude less resources than it is shown in the previous literature.

7 The Attacking Device

From the technological point of view, the attacking device should be equipped with an analog radio frequency module and at least one baseband processor with enough processing power to handle all the concurrent communications occurring during an attack.

The analog module is a standard equipment of every modern mobile phone designed for the medium or high-end market segment, necessary to flexibly process radio signals without specific knowledge of what it is carried on.

Baseband processor, instead, is a critical component because it has to deal with as many different bit streams as the number of ongoing HLR requests. Therefore, many baseband processors may be necessary to ensure the attack effectiveness by generating multiple concurrent attack requests. The race for more and more powerful smartphones makes the availability for this kind of components a non existent problem, furthermore, it is important to notice that according to our study the most limiting bottleneck is the network signaling channels capacity, not the device computational power. Thus, even if the expected evolution will not take place in practice, the implementation of the attacking device will still be possible.

From the control-layer perspective, the actual device requires a significantly simpler design than a traditional UMTS device as it does not need to process all the possible steps in the protocol specification nor to support high bandwidth data transfer demands.

The sketch in Fig. 8 is a prototype of the attacking device that employs one baseband processor for each concurrent

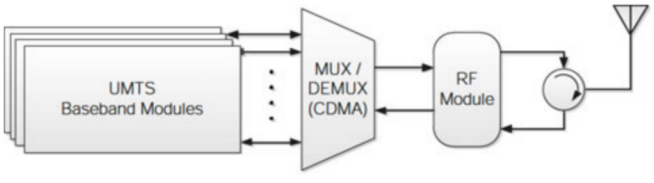

Fig. 8. The attacking device's functional components.

attack request which, in turn, is connected to a WCDMA mux/demux that composes all incoming bit streams to produce a single output signal to be sent to the analog module. Albeit this design is not optimal—since low bandwidth requirements of signaling channels do not allow an efficient exploitation of the whole processing power—it represents a sufficient proof-of-concept scheme.

Only a small part of the protocol should be implemented on the resulting device, 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, the received power of neighbor cells can be computed once and returned whenever asked.

However, the main challenges of the proposed implementation concern the processing power, since all the UMTS interfaces may talk simultaneously and the mux/demux part has to account for all the possible scrambling and orthogonal codes used by each channel in every connection.

8 Conclusions

Several ways to mount DoS attacks against mobile network infrastructures are known. However, in order to make the attack successful, the state-of-the-art attack methodologies [6] are based on GSM network alone and require the availability of botnets with more than 10.000 smartphones with valid SIM modules.

In this work, we have explored a different approach, leveraging the 3G UMTS network and evaluating the possibility to bypass the strict timings enforced by the cellular network protocols by means of radio devices different from the ones available on the consumer market. Accordingly, in order to cope with the above timing limits, we envisioned an ad-hoc attacking device, equipped with multiple UMTS radio interfaces and no SIM modules. This device allowed us to design a novel attack methodology exploiting the network access procedures and to greatly reduce the number of needed resources. Thus, we massively enhanced the threat level of the described attack.

This study first demonstrates that it is possible to inject into the cellular networks signaling traffic without having the control of valid SIM modules. It also shows that the amount of resources that the attacks can squander in the UMTS network infrastructure is sufficient to produce a critical service degradation while reducing the number of needed devices of more than an order of magnitude (from more than eleven thousands to less than five hundred).

Furthermore, we have shown that, by equipping the attacking device with valid SIM cards, it is possible to double the attack effectiveness, reducing the resources needed to just a couple hundred devices. This is a two-order of magnitude effectiveness increment with regards to the previous state of the art.

The feasibility of the attack without a botnet is very important for a two-fold reason: first, the usage of a dedicated device allows gathering the resources needed to mount the attack without interfering with users and running the risk of being discovered before the actual attack; second, avoiding the usage of devices in possession of unaware users allows optimal displacement of attacking equipment and reduces the risk of attack failures due to an incorrect placement of the botnet nodes. In fact, it is possible that an unusual clustering of nodes in a botnet could produce a concentration of devices that saturates the cell signaling bandwidth and prevents some of the nodes to fulfill their full attacking potential.

On the contrary, the device we envision is not owned by a user, and hence conditioned by his movements, so that it can be precisely placed by the attacker and even remotely triggered to start the attack.

All of these factors represent a significant increase in the dangerousness of the attack when compared to the existing ones and can make the described device an interesting target also for the cyber-warfare or cellular network production industry.

References

[1] S. Capkun, M. Cagalj, R. Rengaswamy, I. Tsigkogiannis, J.-P. Hubaux, and M. Srivastava, "Integrity codes: Message integrity protection and authentication over insecure channels," *IEEE Trans. Dependable Secure Comput.*, vol. 5, no. 4, pp. 208–223, Oct.-Dec. 2008.

[2] Y.-L. Huang, F.-Y. Leu, and K.-C. Wei, "A secure communication over wireless environments by using a data connection core," *Math. Comput. Modelling*, vol. 58, no. 5, pp. 1459–1474, 2013.

[3] A. Castiglione, G. Cattaneo, A. De Santis, F. Petagna, and U. Ferraro Petrillo. 2006. "SPEECH: Secure personal end-to-end communication with handheld," in *Proc. ISSE Securing Electronic Busines Processes.* Vieweg, pp. 287–297, [Online]. Available: http://dx.doi.org/10.1007/978-3-8348-9195-2_31

[4] Y.-L. Huang, F.-Y. Leu, I. You, Y.-K. Sun, and C.-C. Chu. (2014). A secure wireless communication system integrating RSA, Diffie-Hellman PKDS, intelligent protection-key chains and a Data Connection Core in a 4G environment. *J. Supercomput.* [Online]. *67(3)*, pp. 635–652. Available: http://dx.doi.org/10.1007/s11227-013-0958-z

[5] B. Blanchet, "A computationally sound mechanized prover for security protocols," *IEEE Trans. Dependable Secure Comput.*, vol. 5, no. 4, pp. 193–207, Oct.-Dec. 2008.

[6] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. La Porta, "On cellular botNets: Measuring the impact of malicious devices on a cellular network core," in *Proc. 16th ACM Conf. Comput. Commun. Security*, 2009, pp. 223–234.

[7] (2013). United States Department of Homeland Security. NIPP 2013: National Infrastructure Protection Plan 2013: Partnering for Critical Infrastructure Security and Resilience. [Online]. Available: http://www.dhs.gov/publication/nipp-2013-partnering-critical-infrastructure-security-and-resilience

[8] (2008). European Commission, European Programme for Critical Infrastructure Protection (EPCIP). [Online]. Available: http://ec.europa.eu/dgs/home-affairs/what-we-do/policies/crisis-and-terrorism/critical-infrastructure/index_en.htm

[9] V. Viduto, C. Maple, and W. Huang, "Managing threats by the use of visualisation techniques," *Int. J. Space-Based Situated Comput.*, vol. 1, no. 2/3, pp. 204–212, 2011.

[10] A. Armando, A. Merlo, M. Migliardi, and L. Verderame, "Breaking and fixing the android launching flow," *Comput. Security*, vol. 39, pp. 104–115, 2013.

[11] A. Mylonas, S. Dritsas, B. Tsoumas, and D. Gritzalis, "Smartphone security evaluation—the malware attack case," in *Proc. Int. Conf. Security Cryptography*, 2011, pp. 25–36.

[12] K. Derr, "Nightmares with mobile devices are just around the corner!" in *Proc. IEEE Int. Conf. Portable Inform. Dev.*, 2007, pp. 1–5.

[13] C. Guo, H. J. Wang, and W. Zhu, "Smart-phone attacks and defenses," in *Proc. 3rd Workshop Hot Topics Netw.*, 2004, pp. 1–6.

[14] P. Traynor, P. McDaniel, and T. La Porta, "On Attack Casuality in Internet-Connected Cellular Networks," in *Proc. 16th USENIX Security Symp.*, 2007, pp. 1–16.

[15] A. Castiglione, R. De Prisco, and A. De Santis. (2009). Do you trust your phone? *Proc. 10th Int. Conf. E-Commerce Web Technol.*, vol. 5692, pp. 50–61 [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03964-5_6
[16] C. Fleizach, M. Liljenstam, P. Johansson, G. M. Voelker, and A. Mehes. (2007). Can you infect me now?: Malware propagation in mobile phone networks. *Proc. ACM Workshop Recurring Malcode*, pp. 61–68. [Online]. Available: http://doi.acm.org/10.1145/1314389.1314402
[17] C. Mulliner and J.-P. Seifert, "Rise of the iBots: Owning a telco network," in *Proc. 5th Int. conf. Malicious Unwanted Softw.*, 2010, pp. 71–80.
[18] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner. (2011). A survey of mobile malware in the wild. *Proc. 1st ACM Workshop Security Privacy in Smartphones Mobile Dev.*, pp. 3–14. [Online]. Available: http://doi.acm.org/10.1145/2046614.2046618
[19] P. Traynor, W. Enck, P. McDaniel, and T. La Porta, "Mitigating attacks on open functionality in SMS-capable cellular networks," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw.*, 2006, pp. 182–193.
[20] M. Khan, A. Ahmed, and A. R. Cheema, "Vulnerabilities of UMTS access domain security architecture," in *Proc. IEEE 9th Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput.*, 2008, pp. 350–355.
[21] G. Kambourakis, C. Kolias, S. Gritzalis, and J. Hyuk-Park, "Signaling-oriented DoS attacks in UMTS networks," in *Proc. 3rd Int. Conf. Workshops Adv. Inform. Security Assurance*, 2009, pp. 280–289.
[22] N. Gobbo, A. Merlo, and M. Migliardi. (2013). A denial of service attack to GSM networks via attach procedure, *Proc. ARES Workshop*, vol. 8128, pp. 361–376, [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40588-4_25
[23] 3GPP, (2012). *TS 25.214—Physical layer procedures (FDD)*. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/25214.htm
[24] 3GPP, (2012). *TS 24.008—Mobile radio interface Layer 3 specification; Core network protocols; Stage 3*. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/24008.htm
[25] 3GPP, (2012). *TS 25.322—Radio Link Control (RLC) protocol specification*. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/25322.htm
[26] C. Johnson, H. Holma, and I. Sharp, "Connection setup delay for packet switched services," in *Proc. 6th IEEE Int. Conf. 3G Beyond*, 2005, pp. 1–5.
[27] H. Holma and A. Toskala, *WCDMA for UMTS*. Hoboken, NJ, USA: Wiley, 2002.

Alessio Merlo received the MSc degree in computer science from the University of Genova, in 2005 and the PhD degree in computer science from the University of Genova, Italy, in 2010, where he worked on performance and access control issues related to Grid Computing. He is currently serving as an assistant professor at E-Campus University, Novedrate, Italy, and as an associated researcher at Artificial Intelligence Laboratory (AILab) at DIBRIS, University of Genova. His currently research interests include performance and security issues related to web, distributed systems (Grid, Cloud), and mobile (Android platform). He is involved as a member in program committees of international conferences (IFIP-SEC, AINA, ARES, HPCS, . . .) and in the editorial board of an international journal (*Journal of High Speed Networks*).

Mauro Migliardi received the PhD degree in computer engineering in 1995. He was a research associate and an assistant professor at the University of Genoa and a research associate at Emory University as Co-PI in the HARNESS heterogeneous metacomputing project. Currently he is an associate professor at the University of Padua and a supply professor at the University of Genoa. He is also a member of the Scientific Committee of the Center for Computing Platforms Engineering and he has received the 2013 Canada-Italy Innovation Reward. His main research interest includes distributed systems engineering in general; recently he focused on mobile systems, human memory support services, energy awareness, and green security. He has tutored more than 80 among the bachelor, master, and PhD students at the Universities of Genoa, Padua, and Emory, and he has authored or co-authored more than 100 scientific papers published in national and international, peer reviewed conferences, books and journals.

Nicola Gobbo received the Laurea degree in computer engineering from the University of Padua. In the past, he has been a consultant for the RSA, The Security Division of EMC, Telecom Italia, and he is now a security consultant for reply in the Milan Area. His main expertise is about network and mobile security.

Francesco Palmieri received the MS degree and the PhD degree in computer science from the Salerno University. He is an assistant professor at the Second University of Napoli. His research interests include advanced networking protocols and architectures and network security. He has been the director of the Networking Division of the Federico II University of Napoli and contributed to the development of the Internet in Italy as a senior member of the Technical-Scientific Advisory Committee and of the CSIRT of the Italian NREN GARR. He serves as the editor-in-chief of an international journal and participates to the editorial board of other ones.

Aniello Castiglione (S'04, M'08) received the degree in computer science and the PhD degree in computer science from the same university. He joined the Department of Computer Science of the University of Salerno in 2006. He serves as a reviewer for several international journals and has been a member of international conference committees. He has been involved in forensic investigations, collaborating with several law enforcement agencies as a consultant. His research interests include security, communication networks, information forensics, security, and cryptography. He is a member of various associations, including the IEEE and the ACM.