usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

# Let Me Answer That For You: Exploiting Broadcast Information in Cellular Networks

Nico Golde, Kévin Redon, and Jean-Pierre Seifert,
*Technische Universität Berlin and Deutsche Telekom Innovation Laboratories*

This paper is included in the Proceedings of the
22nd USENIX Security Symposium.

August 14–16, 2013 • Washington, D.C., USA

ISBN 978-1-931971-03-4

# Let Me Answer That For You: Exploiting Broadcast Information in Cellular Networks

Nico Golde, Kévin Redon, Jean-Pierre Seifert
*Technische Universität Berlin and Deutsche Telekom Innovation Laboratories*
{*nico, kredon, jpseifert*}@*sec.t-labs.tu-berlin.de*

## Abstract

Mobile telecommunication has become an important part of our daily lives. Yet, industry standards such as GSM often exclude scenarios with active attackers. Devices participating in communication are seen as trusted and non-malicious. By implementing our own baseband firmware based on OsmocomBB, we violate this trust and are able to evaluate the impact of a rogue device with regard to the usage of broadcast information. Through our analysis we show two new attacks based on the paging procedure used in cellular networks. We demonstrate that for at least GSM, it is feasible to hijack the transmission of mobile terminated services such as calls, perform targeted denial of service attacks against single subscribers and as well against large geographical regions within a metropolitan area.

## 1 Introduction

While past research on *Global System for Mobile Communications (GSM)* mainly focused on theoretical research [17, 18], a very recent research direction challenged the fundamental GSM security assumptions with respect to the practical availability of *open* GSM equipment. The assumptions have been made on both sides of the radio part of the cellular network. One side of the radio link is the *Base Station System (BSS)* consisting of the *Base Transceiver Station (BTS)* and the *Base Station Controller (BSC)*, while the other side of the radio part is the modem or the so-called baseband of a cellular phone. Traditionally, both radio stacks have been carefully kept out of reach for any kind of malicious activities.

But a booming market for used telecommunication equipment, cheap software defined radios, leakage of some hardware specifications, and a well-trained open source community finally broke up this closed cellular world. The overall community work culminated in three open source projects: OpenBSC, OpenBTS, and OsmocomBB [20, 25, 45]. These open source projects constitute the long sought and yet *publicly available* counterparts of the *previously closed* radio stacks. Although all of them are still constrained to 2G network handling, recent research provides open source software to tamper with certain 3G base stations [24]. Needless to say that those projects initiated a whole new class of so far unconsidered and practical security investigations within the cellular communication research, [28, 30, 34].

Despite the recent roll-out of 4G networks, GSM remains the dominant cellular standard in many countries. Moreover, as most new LTE devices are backwards compatible to GSM, this older standard will not vanish soon at all, but rather complement 3G and LTE connectivity in areas with pure GSM coverage. Several other reasons such as worse indoor coverage and the lower number of deployed UMTS and LTE base stations contribute to this. Additionally, telecommunication providers have already begun to reuse their existing GSM infrastructure within non-voice scenarios which require a much slower data communication than modern network technologies are capable of. This is especially the case for *Machine to Machine (M2M)* or so-called *Internet of Things (IoT)* communications over GSM. Corresponding applications will soon become parts of our daily life and will make us more dependent than ever on GSM, cf. [19, 35]. Given this pervasive GSM usage, it is very important to evaluate the security offered by a standard which is more than 20 years old and is based on assumptions, many of which no longer hold true.

This paper continues the challenge of the mobile security assumption that *certain active attacks can be safely excluded* from the threat model. Towards this goal we show novel attacks against mobile terminated services. While the root cause also exists in newer standards such as UMTS or LTE, we demonstrate the impact of it in commercially deployed GSM networks. To the best of our knowledge, the limitations of currently available hard- and software would make it very difficult

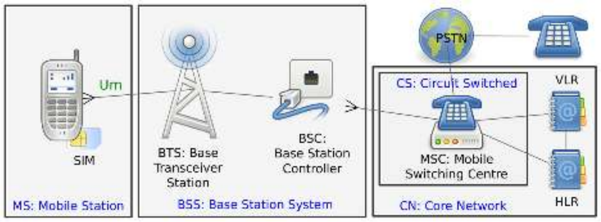


Figure 1: Simplified GSM network infrastructure.

to test these attacks in UMTS and LTE networks. Prior to publishing this research, we responsibly notified the respective standard organisations via a carrier of our research results.

In summary, we make the following main contributions:

- We present the paging response attack, a novel and practical attack against mobile terminated services.
- We show the feasibility and the implementation of a mobile phone firmware which is capable to steal a short message over-the-air and to perform denial of service attacks against mobile terminated services in GSM networks. Furthermore, we evaluated these attacks to be present in major European operator networks.
- We eventually assess the boundary conditions for a large-scale paging response attack in order to cause denial of service conditions within a large geographical area of a major city.

The remainder of the paper is structured as follows. Section 2 provides an overview of the 3GPP GSM network infrastructure, as well as details about logical channels and paging protocol procedures required to understand our attacks; Section 3 details our novel attack that exploits the paging procedure as used in GSM; Section 4 describes characteristics of location areas in a large metropolitan area and the respective requirements to perform a large-scale denial of service attack against these regions; Section 5 discusses two different countermeasures to address the attacks; Section 6 provides an overview of related research; Section 7 concludes our research.

## 2 Background and Overview

This section briefly describes the GSM cellular network infrastructure. We continue to explain the important types and functions of logical channels. Furthermore, we depict the protocol details required to understand the basis of our attack.

### 2.1 GSM Infrastructure

Despite the complexity of a complete GSM mobile network architecture [3], only a few entities are relevant to this work. In the following paragraph, we provide the necessary background on the infrastructure components of relevance to this research. Figure 1 illustrates the architecture and connections between these components:

- *BTS*: The Base Transceiver Station is a phone's access point to the network. It relays radio traffic to and from the mobile network and provides access to the network over-the-air. A set of BTSs is controlled by a Base Station Controller (BSC) and is part of a Base Station System (BSS).
- *MS*: The Mobile Station is the mobile device interacting with the mobile operator network. It comprises hardware and software required for mobile communication (baseband processor, SIM card, and a GSM stack implementation). The MS interacts with the BTS over a radio link, also known as the $U_m$ interface. In this paper, the mobile phone of a victim is often referred to as MS. We will also use the term MS, user, subscriber, phone, and mobile device interchangeably.
- *MSC*: The Mobile Switching Center [6] is a core network entity responsible for routing services, such as calls and short messages, through the network. It utilizes components from BSSs to establish connections to mobile devices, organizes hand-over procedures and connects the cellular network to the Public Switched Telephone Network (PSTN).
- *VLR*: The Visitor Location Register maintains location and management data for mobile subscribers roaming in a specific geographical area handled by an MSC. It acts as a local database cache for various subscriber information obtained from the central Home Location Register (HLR), e.g., the mobile identity. A subscriber can only be present in one VLR at a time. Each of the areas served has an associated unique identifier, the Location Area Code (LAC) [3,8]. As soon as a phone leaves a certain geographical area called *Location Area (LA)*, it has to perform the Location Update procedure [4] to notify the network of this event.

### 2.2 GSM Logical Channels

The available GSM frequencies are shared among a number of mobile carriers. Each of the GSM frequency bands is divided into multiple carrier frequencies by means of Frequency Division Multiple Access (FDMA). A BTS serves at least one associated carrier frequencies identified by the Absolute Radio-Frequency Channel Number (ARFCN). The ARFCN provides a dedicated pair of uplink and downlink frequencies for receiving and transmitting data over the $U_m$ interface [10]. Because the radio frequency is shared among a number of subscribers, GSM uses Time Division Multiple Access (TDMA) as channel access method and divides physical channels provided by the ARFCN into 8 time slots. A sequence of 8 consecutive time slots is called a TDMA frame. Multiple TDMA frames form a multiframe. It consists either of 51 or 21 TDMA frames (respectively control frames or traffic frames). Multiframes are further partitioned to provide logical channels.

The two categories of logical channels in GSM are *control channels* and *traffic channels* [5]. Control channels provide means for signaling between the network and the MS. Because our attack is solely based on signaling, we focus on the details of control channels. There are three categories of control channels:

- *BCH*: Broadcast Channels provide a point-to-multipoint, unidirectional channel from the BTS to mobile stations (transmitted on the downlink frequency). Among other functionalities, they act as beacon channels and include logical channels for frequency correction (FCCH), synchronization (SCH), and information about the cell configuration and identity (BCCH) [5, 7].

- *CCCH*: Common Control Channels are used for signaling between the BTS and MS, both on the uplink and downlink. They are used by the MS to request radio resources and to access the mobile network.

- *DCCH*: Dedicated Control Channels carry signaling messages related to handover procedures or connection establishment, e.g., during call setups.

For our attack, we are mainly interested in logical channels that are part of the CCCH and DCCH categories. These categories consist of several logical channels. The logical channels of interest are as follows:

- *PCH*: The Paging Channel is used by the BTS to inform an MS about an incoming service (via paging request messages on the downlink channel). The PCH, which is part of the CCCH, will be monitored by any MS in idle mode unless it is currently using a dedicated channel.

- *RACH*: The Random Access Channel provides a shared uplink channel utilized by the MS to request a dedicated channel from the BTS. Placing a phone call or receiving an incoming service always requires a phone to setup a dedicated signaling channel beforehand.

- *AGCH*: The Access Grant Channel provides a downlink channel used by the BTS to transmit assignment messages that notify mobile stations of assigned channel details. A successful channel request on the RACH will result in an Immediate Assignment message on the AGCH. These assignment messages contain the required configuration parameters that enable the MS to tune to the requested channel.

- *SDCCH*: The Standalone Dedicated Control Channel is used on both uplink and downlink. It is employed for call setup and signaling between BTS and MS. Furthermore, it can be utilized to transmit short messages to the MS.

It is important to note that both the BCH and CCCH channel types are point-to-multipoint channels. This implies that information on the logical downlink channels is broadcasted to all subscribers served by a specific BTS. Throughout this work we will see how this can be abused to model new attacks.

## 2.3 Mobile Terminated Service Procedures

The GSM specifications differ between traffic originating or terminating at a mobile phone. This is referred to as Mobile Originated (MO) and Mobile Terminated (MT) traffic. As outlined previously, we aim to attack MT services, such as phone calls or SMS. Thus, in the following we concentrate on the underlying protocol procedures associated with MT services [4].

In order to deliver a service to a phone, the MSC needs to determine the location of the respective subscriber. This has to be done for two reasons. First, mobile phones will be idle most of the time to save battery power and so will not be in constant contact with the network. Thus, the operator does not always know the specific BTS that provides the best reception level to the MS. Therefore, it must broadcast this signal of an incoming service through at least the entire location area. Second, broadcasting this information through the whole operator network would impose a huge performance overhead and possibly overload the paging channel [42].

In a first step, the core network determines the responsible MSC/VLR for the target subscriber with the help of the HLR. Next, the MSC obtains the location information for the destination subscriber from the VLR and sends a *paging message* to all BSCs in the subscriber's location area. This message includes a list of cell identifiers/base stations serving the specific location area [13]. The message also contains the mobile identity of the subscriber, which is usually either a *International Mobile Subscriber Identity* (IMSI) or a *Temporary Mobile Subscriber Identity* (TMSI). We illustrate the remaining protocol logic using a successful MT phone call as depicted in Figure 2.

1. The BSC sends a *paging command* message which includes the subscriber identity to all base stations within the location area. All base stations re-encapsulate the mobile identity and transmit it as part of a *paging request* message on the downlink PCH.

2. When receiving a paging request on the PCH, each MS compares the Mobile Identity (MI) included in the request with its own. The result determines whether the message is addressed to itself or a different subscriber.

3. In case of an identity match, the MS needs to acquire access to Radio Resources (RR) in order to receive the MT service. To do so, it sends a *channel request* including a random reference number on the uplink RACH.

4. Upon receipt of the channel request, the network allocates radio resources and a dedicated channel. Next, it acknowledges the request and sends details of the allocated channel to the MS in an *immediate assignment* message on the AGCH downlink. To allow the MS to identify its assignment, the message contains the random reference of the requester.

5. The AGCH is a shared downlink channel. Therefore, an MS receiving an assignment message compares the included reference with the one sent in the request. If the reference matches, the MS tunes to the dedicated signaling channel included in the assignment.

6. After this step succeeded, the Mobile Station establishes a signaling link, usually over the SDCCH, by sending a GSM Layer 2 *SABM* frame containing a Layer 3 *paging response* message.

7. Following this, the MS and BTS undergo an authentication, ciphering and service setup procedure. Details of this procedure are not relevant for our attack. We skip these details here.

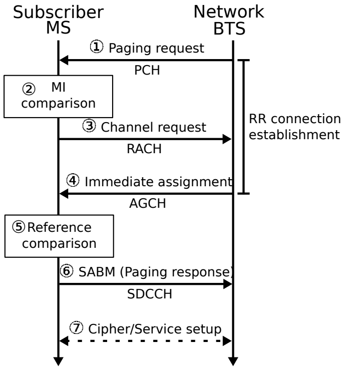


Figure 2: Mobile Terminated (MT) paging procedure.

The GSM standard specifies [4] three types of paging requests – type 1, 2, and 3. The type stipulates the number of subscribers that can be addressed with the paging request. Type 1 can page one or two subscribers, type 2 two or three subscribers, and type 3 paging requests are directed towards four subscribers at once. A recent study [30] suggests that in real operator networks the vast majority of paging requests is of type 1. During our experiments, we verified that 98% of all paging requests that we observed are type 1 requests. Therefore, we ignore type 2 and type 3 paging requests in our study.

## 3 Attack Description

In this section, we will provide the theoretical background of our attack, introduce our experimental setup and elaborate on the feasibility of such an attack.

### 3.1 The Two Threat Models

**Denial of Service Attacks.** The first threat comprises an active attacker, interested in significantly *disturbing* mobile terminated services within a specific geographical area, e.g., a district or a part of a city. In certain situations it is desirable to ensure that a person or a device is not reachable via mobile telephony. For example a third-party may want to prevent a specific call from reaching the victim. The effect would be similar to the ability of selectively jamming incoming services for a set of subscribers. This includes individuals and groups of individuals. Such an attack would also have considerable business ramifications. While it would not compromise the general operation of the carrier, it would affect their revenue. The inability to receive a phone call will not only leave angry customers, it further impacts the generated billing as subscribers are charged when a call is connected. If any subscriber is able to place phone calls, but nobody is able to receive services, no profit is created. An exception here are short messages, as SMS operates in store-and-forward fashion and does not create billing on delivery of a message, but on its submission.

**Mobile Terminated Impersonation.** The second threat considers an attacker who aims to *hijack* a mobile terminated service. As a result, the service would be delivered to the attacker instead of the victim. This turns a passive adversary, who is able to observer air traffic, into an active attacker who can accept the mobile terminated service and impersonate the victim. For example an attacker could be interested in hijacking the delivery of an SMS message. Consequently, it is possible to read its content and at the same time prevent its submission to the victim. In practice this could, for example, allow an attacker to steal a mobile TAN (mTAN), which is often used as two-factor authentication for online banking, or any other valuable secret from the message. We also consider an attacker who wants to impersonate a victim that is being called. By hijacking the MT call setup, it is almost impossible for the calling person to verify the callee's identity by means other than the voice.

## 3.2 Paging Response Attack Description

Our attack is inspired by two specific properties of GSM networks and its protocols.

**Network State:** GSM networks involve complex state machines [4] and face high amounts of traffic while operating on tight radio resource constraints. Consequently, it is desirable to keep states as short as possible.

**Broadcast Information:** the paging procedure is initiated on a broadcast medium, namely the PCH portion of the CCCH, and more importantly is performed before any authentication or cipher setup takes place. This implies that any subscriber, including an adversary phone, is able to observe paging requests for other subscribers, plus the inherent inability of the network to distinguish between a fake paging response and a genuine one.

As a net result, it is possible to exploit these aspects to send paging response messages on behalf of a victim being paged. The network stack can under no circumstances determine which of the replies is the legitimate paging response by the intended subscriber.

**Denial of Service.** The GSM documents do not specify the network behavior in such a situation. Therefore, the behavior of such a race condition is implementation dependent and may be exploitable. However, the state machine nature of GSM protocols suggest that if an attacker is able to answer a paging request faster than the intended subscriber, it will no longer be in a state in which it expects a paging response and thus will ignore the message of a victim. Consequently, the victim will receive a channel release message from the network. Next, the service setup will not succeed if the attacker does not provide the correct cryptographic keys required to complete authentication and cipher setup. Accordingly, the service setup cannot proceed and for example, a call will be dropped. The result is a novel and powerful denial of service attack against MT services that 1. does not rely on frequency jamming; 2. does not rely on resource exhaustion; and 3. is very hard to detect.

We verified that it is indeed possible to win the race for the fastest paging response time, as we will demonstrate. We were able to carry out such an attack in all major German operator networks including O2, Vodafone, T-Mobile, and E-Plus.

**MT Session Hijacking.** Exploiting the paging procedure does not only allow to disturb communication. It is important to note that in certain network configurations, this attack could be abused beyond performing denial of service attacks. Not all countries properly authenticate each service and use encryption. For example, only under 20% of the networks analyzed by the gsmmap project [41] authenticate mobile terminated phone calls 100% of the time. 50% of the tested networks only authenticate 10% of the services [28].

In such a network, an adversary can effectively takeover any MT service that is not authenticated and impersonate a victim. We assume a network without encryption and insufficient authentication as above. If the attacker is able to successfully exploit the race condition on the air interface, it is possible to directly hijack an MT service by following the protocol specifications. The paging response attack proceeds as in the DoS scenario. However, in this case, by winning the race, an attacker can accept, e.g., a victim's phone call or short message.

The victim of such an attack is thus faced with two consequences. For a mobile terminated call, it is not safe to assume that the called party is indeed the desired person. For short messages this implies that a message may not reach the victim, but additionally also that its contents cannot be considered secret.

Even if the network is configured to use encryption, an attacker is merely required to perform an additional step. In an encrypted network without proper authentication, the paging procedure is followed by the cipher setup. During this process to create an encrypted channel, the network sends a *cipher mode command* message to notify the MS of the encryption algorithm to be used. The *cipher mode complete* response from the MS indicates a completion of the cipher setup. In a network that uses encryption, this response has to be encrypted using the session key $K_c$ as input to the A5 encryption algorithm. This session key is derived from a secret key $K_i$ that is stored on the SIM card issued by the operator and a random challenge *RAND* sent from the network to the MS. Due to the lack of perpetual authentication, an attacker can fully impersonate the victim after cracking the session key $K_c$ and sending the *cipher mode complete* message. The cracked session key then allows to decrypt the subsequent communication that follows the cipher setup.

In practice, essentially both commonly used GSM cipher algorithms, A5/2 and A5/1, have been broken and demonstrated to be cryptographically weak [17, 18, 23, 39]. The session key can be acquired before hijacking the service by sniffing air traffic and using the kraken tool [40]. Also, some networks are configured to still use A5/0 [26], which does not provide any encryption. This further simplifies such an attack in those commercially deployed networks. Furthermore, for the subsequent paging response attack, an attacker does not even require physical proximity to a victim, because, as explained earlier, the carrier network is paging throughout an entire location area. In order to exploit this, an attacker requires a mobile device that enables him to observe traffic on the air interface and send arbitrary messages to the network. Additionally, a practical attack requires the fake response to arrive prior to the victim's message. Therefore, the attack is significantly challenging in terms of timing.

We successfully implemented both, the MT service hijacking and the denial of service attack. For the sake of simplicity, we obtained the session key through the SIM browser in the engineering mode of a Blackberry phone. Nevertheless, as outlined before this step, it can be trivially obtained by a 3rd party by using a tool like kraken [40]. Cracking of $K_c$ is merely a step that has to be performed prior to our attack, but is not part of the problem itself, which is the race condition. Given a known $K_c$, our code to take over an MT session, can hijack the transmission of a short message delivery in a real network.

It is important to note that the main reason for evaluating the paging race condition in GSM was the availability of freely modifiable hardware and software. However, modern telecommunication standards such as UMTS or LTE are making use of exactly the same paging procedure principles [11,14,15]. Insufficient cryptography and authentication further escalate the problem, but the root cause does not only pertain to GSM.

We will continue to examine the requirements, boundary conditions, and feasibility of mounting such an attack in practice.

## 3.3 Experimental Setup

Launching such an attack requires hardware and software to interact with GSM base stations. More precisely, the attack relies on a device which allows us to modify its baseband (BB) implementation in order to control its radio communication. Traditionally this has been very difficult due to the closed nature of the GSM industry (phone manufacturers, baseband vendors, infrastructure equipment suppliers). For many years there existed no freely modifiable radio communication hardware with GSM stack implementations. While the GSM specifications are publicly available (very comprehensive though, over 1000 PDF documents), there are very few manufacturers of GSM equipment who have released any public documentation.

However, this situation has changed in the last years with the availability of inexpensive hardware such as the Universal Software Radio Peripheral (USRP) [22] and various software implementations around the Osmocom [45] project. Additionally, in 2004 the source code of the Vitelcom TSM30 mobile phone was uploaded to a Sourceforge project [37] which allowed a broader audience to study a GSM phone stack for the first time.

**Hardware Selection.** There are basically three possible choices when it comes to the hardware selection of our desired radio device: *USRP*, *Vitelcom TSM30*, and certain *TI Calypso chipset based phones*. All of these devices can be utilized as GSM radio transceivers with software modifications. Yet some of these come with intrinsic disadvantages. First, for the USRP there is currently no GSM baseband implementation that allows the device to be used as a handset. While we could have implemented this, it would have been a very demanding task. Second, even though available, the TSM30 source code is a full-featured baseband implementation, which is too complex for our needs. Moreover, the availability of TSM30 devices is sparse and they are not easy to obtain.

Instead we used Motorola C123 and Motorola C118 phones, which are based on the TI Calypso chipset. These phones are inexpensive (around 20 Euros), easy to obtain in quantity, and more importantly can be used in combination with the Free Software baseband implementation OsmocomBB [47]. This enables us to receive over-the-air traffic and send arbitrary GSM frames.

**Implementation.** OsmocomBB implements a simplified version of the GSM stack. The GSM physical layer (L1) firmware runs on the phone, while the data-link layer (L2) and Layer 3 (L3) run on a computer as an application (layer23). L1 and layer23 communicate with each other via a UART serial connection. Layer 2 implements a modified version of the Link Access Protocol for the D channel (LAPD) used in ISDN, called Link Access Protocol on the Dm channel (LAPDm). Layer 3 comprises three sublayers: Radio Resource management, Mobility Management, and Connection Management. As our attack is based on paging, which is part of Layer 3, we required a modified version of the layer23 application.

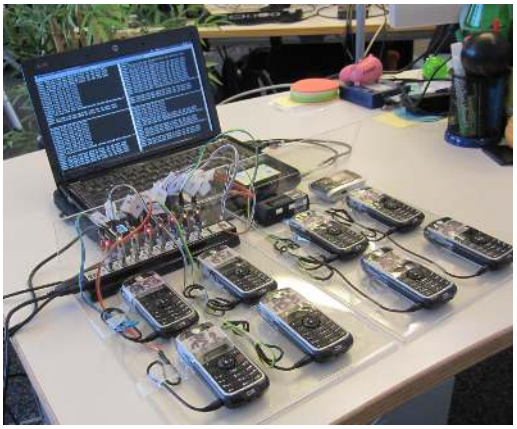


Figure 3: Experimental setup: Motorola C1XX phones with custom firmware, GPS receivers, and a laptop for serial communication.

In practice our attack is particularly time critical, because we have to win a race condition on the air interface. It became evident that a layer23 implementation that runs on a computer is far too slow to win the race given the bottlenecks such as queueing between multiple layers, scheduling, and the use of UART serial communication. Consequently, we reimplemented a minimal version of LAPDm and Layer 3 directly in the L1 firmware to allow it to run solely on the phone. Specifically this includes the paging protocol, which is part of the radio resource sublayer.

Figure 3 shows our experimental setup consisting of a notebook and several OsmocomBB phones. The serial cables are required in order to flash the firmware. Using this implementation we can camp on specific ARFCNs, observe paging requests within a location area, and send arbitrary GSM layer2/layer3 messages in a timely manner. Additionally, we used OpenBTS [20] in combination with a USRP as a BTS to test our setup and perform various measurements as later described in Section 3.5.

## 3.4 Targeted Attacks

Attacking individual persons requires our OsmocomBB phone to observe air traffic and respond to specific paging requests. In particular paging requests that contain the victims mobile identity. For privacy reasons, most network operators use TMSIs as mobile identities rather than the static IMSI. The TMSI is only valid within a location area and is subject to frequent changes [9]. Therefore, we need to determine the presence and the TMSI of a victim in a given location area.

For this we implemented the method proposed by Kune et al. to reveal the mapping between TMSI and subscriber [30]. We modified OsmocomBB's layer23 mobile application and introduced functionality that issues *n* (where *n* is 10-20) phone calls in a row. Next, the application terminates the connection before the target phone is ringing, but late enough so that the network generates a paging request. The victim phone does not ring during this early stage of the protocol flow, because it does not know yet what type of service is incoming. In our tests we empirically determined that, e.g., a time of 3.7 seconds after the *CC-Establishment confirmed* state has the desired effect in the O2 network. The exact timing may differ slightly, depending on the network that is used to initiate the call and the network in which the victim resides.

At the same time, a second phone is monitoring the PCH of any BTS within the target location area for paging requests. All TMSIs contained in the observed paging requests are logged together with a precise timestamp of the event. It makes sense to choose the ARFCN with the best signal reception to minimize errors and possible delays. By first limiting the resulting log to time ranges in which our calls were initiated, we can extract a number of candidate TMSIs. Further filtering the result set for TMSIs occurring in repeating patterns that reflect our call pattern yields to a very small set of candidate TMSIs or even single TMSIs. This process can be repeated to narrow down the set of candidate TMSIs to a manageable number. If the network uses IMSIs for identification, then an attacker could use the same process to determine the subscriber's identity. Alternatively, an attacker could use a Home Location Register query service to obtain the IMSI directly [1].

By default, the monitoring phone does not react to any paging request. After obtaining the victim TMSI, we transfer the TMSI via HDLC over the serial connection to the monitoring phone. This also changes the phone's role from a solely passive listener to an attacker. It starts to compare TMSIs contained in paging request with the supplied victim TMSI. On every match, the attacking phone promptly initiates the previously introduced paging protocol procedure to respond first. As a result, the paging response by the victim will be ignored and the call will be dropped unless we fully accept the service. At this point, it is not possible to reach the victim anymore. To block MT services over a longer period of time, the subscriber identification procedure needs to be reissued due to TMSI reallocations over time [4].

## 3.5 Feasibility

The success of such an exploit depends essentially on the response time of the attacker and victim devices. To achieve maximum impact, an attacker phone needs to respond faster than the "average" customer device. The response time of the phone depends on a number of fac-

Table 1: List of tested phones, baseband chipset, and baseband vendor.

| Phone model | BB chipset | BB vendor |
|---|---|---|
| Blackberry Curve 9300 | Marvell PXA930 | Marvell |
| iPhone 4s | MDM6610 | Qualcomm |
| Samsung Galaxy S2 | XMM 6260 | Infineon |
| Nokia N900 | TI Rapuyama | Nokia |
| Nokia 3310 | TI MAD2WDI | Nokia |
| Motorola C123 | TI Calypso | OsmocomBB[1] |
| SciPhone Dream G2 | MT6235 | Mediatek |
| Sony Ericsson W800i | DB2010 | Ericsson |
| Sony Xperia U | NovaThor U8500 | ST-Ericsson |

[1] Layer1 paging attack code and modified layer23 application.

tors that are difficult to measure. This includes signal quality, weather, network saturation, application processor operating system, GSM time slots, and others. Yet, most of these parameters only have very little impact on the overall response time.

As the baseband chipset and its GSM stack implementation handles all radio communication, including the upper layer GSM logic, we suspect it to be a key contributor to a fast response time. We validate this claim by measuring the timing of various phones with different baseband vendors. Referring to a market report [2], Qualcomm and Intel alone account for 60% of the baseband revenue in 2011. Yet, relevant baseband chips and stacks that are currently available in mobile phones on the market are Qualcomm, Intel (formerly Infineon), Texas Instruments, ST-Ericsson, Renesas (formerly Nokia), Marvell, and Mediatek. We tested timing behavior for different phones for each of these vendors. Additionally, we also tested the response time for the OsmocomBB layer23 application to back up our claim that this implementation is too slow to perform our attack. Table 1 lists the tested phone models, chipset names, and the corresponding baseband vendor.

**Timing Measurements.** It is not feasible to modify the tested devices itself for measurements, as we only have access to the operating system on the application processor, and not the baseband. Furthermore, the phone could only guess when its response hits the serving network. Thus, in order to estimate the paging response time, we operate our own test GSM BTS based on a USRP and OpenBTS [20]. OpenBTS implements a simplified GSM network stack running on commodity hardware while using the USRP device as a transceiver. We patched OpenBTS to obtain timing information for the different steps during the paging procedure. Specifically, we are interested in the time a phone needs to acquire a radio channel and to send the paging response. This includes two parts of the paging procedure, the time between the initial paging request and the channel request, and the time between the initial paging request and the reception of the paging response. We log both of these timestamps for the relevant baseband vendors in nanoseconds using *clock_gettime(2)*. Additionally, we measure the same for an attack phone running our own lightweight, OsmocomBB-based baseband implementation. To trigger paging activity, we consecutively send 250 short messages, one after each channel teardown, to our test devices.

While we could have also used software like OpenBSC [25] in combination with a nanoBTS [27], we decided to utilize OpenBTS to be in full control over the transmission and reception. The nanoBTS is controlled over Ethernet, runs its own operating system, including scheduling algorithms, and cannot be modified. Thus, we used OpenBTS to minimize the deviation that may occur due to the nature of this BTS device.

**Timing Observations.** Figure 4 summarizes the results of our time measurements for each baseband vendor. It shows the elapsed time between the first paging request message sent to the phone, the arrival of the channel request message, and the occurrence of the paging response. Interestingly, the generation of the phone had little influence on the response timing. In our tests, a Nokia 3310, which is almost 10 years older than the tested Nokia N900, shows almost the same timing behavior. We do not have a definitive answer to explain this observation. However, a plausible explanation can be found in the age of GSM. GSM was developed in the 1980s and most of the mobile telephony stacks for GSM are of this

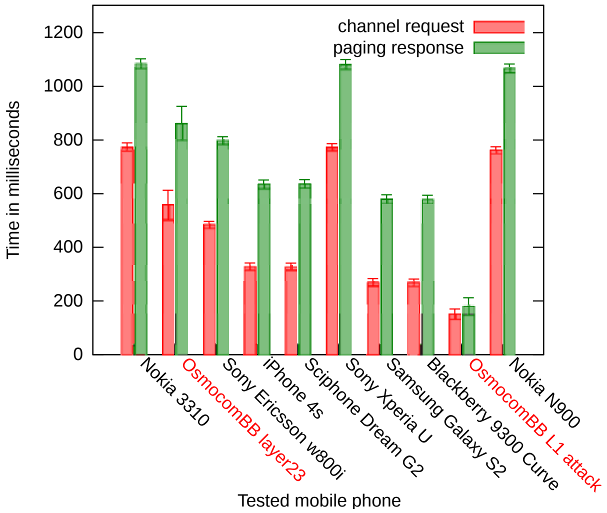


Figure 4: Time difference between initial paging request and subsequent channel request or paging response for different baseband vendors. Confidence interval: 95%.

era. As most baseband vendors nowadays concentrate their efforts on exploring the technical challenges of 3G and 4G telephony standards, we believe that GSM stacks have not been modified for a long time. We do not expect significant modifications of baseband stacks by the respective vendors nowadays. Thus, we assume that timing behavior across different phone platforms using the same baseband will show similar patterns.

The most important observation from Figure 4 is that on average, with a confidence interval of 95%, our minimal OsmocomBB-based implementation is the fastest in transmitting the channel request and paging response. For our implementation, there is roughly a 180 milliseconds delay between the paging request and the arrival of the paging response. Thus, on average our attack implementation is able to transmit the final paging response prior to all other major basebands and can be conducted within the duration of a single multiframe (235.4 ms). This includes the OsmocomBB layer23 mobile application, which is significantly slower than our self-contained layer1 attack software and shows similar timing performance as conventional phones.

Therefore, with a very high likelihood, our software is able to win the race. It is also noteworthy that our lightweight stack can transmit the paging response almost immediately after the channel request (and reception of the Immediate Assignment). The test devices show a gap of at least 200ms before the transmission of the paging response. We expect that this is related to internal scheduling algorithms and queuing mechanisms between different layers of the baseband implementation.

## 4 Attacking Location Areas

Besides attacking individual subscribers, we show that it is also possible to leverage this attack to disrupt network service in large geographical regions. As explained in Section 2, the serving network does not always have the knowledge of the exact location a subscriber resides in. As a consequence, it also does not know which BTS is currently within a good reception of the mobile device. The phone announces a change of the location area by performing the *Location Update* [4] procedure. By monitoring *System Information* [4] messages on the Broadcast Control Channel (BCCH), a phone can keep track of location areas served by the BTSs within reception. The aforementioned lack of knowledge is compensated by the network by distributing paging requests throughout all base stations in the location area. This implies that an adversary is able to observe and respond to paging requests not only transmitted by a single a BTS, but within a larger geographical region formed by the location area.

We already showed in Section 3.5 that we win the race for the paging response with high probability. Given that an attacker is able to answer all paging requests that can be observed on the PCH, it is possible to perform a denial of service attack against all MT services within the location area. Depending on its size, the impact of this would be massive, e.g., breaking MT calls in areas as large as city districts or even bigger regions. However, in practice there are a few obstacles to consider.

Depending on the paging activity, it is unlikely that service in an entire geographical can be disrupted by a single attacker phone. In order to send the paging response, the MS has to tune to a dedicated channel. As a result, it would not be able to observe paging requests while being in dedicated mode. After sending the response, the attacker MS has to resynchronize with the BTS to observe CCCH/PCH traffic again. By logging timestamps for the various protocol steps, we measured the time for this procedure on the OsmocomBB side. On average we need 745 milliseconds to resynchronize in order to receive further paging requests after we sent the response. Furthermore, as shown in Figure 4, we need on average 180 milliseconds to transmit the paging response. This means that in ideal conditions, with a single phone, we are able to handle up to

$$\frac{60\,\mathrm{s}}{745\,\mathrm{ms}+180\,\mathrm{ms}} = 64.8 \text{ paging requests per minute.}$$

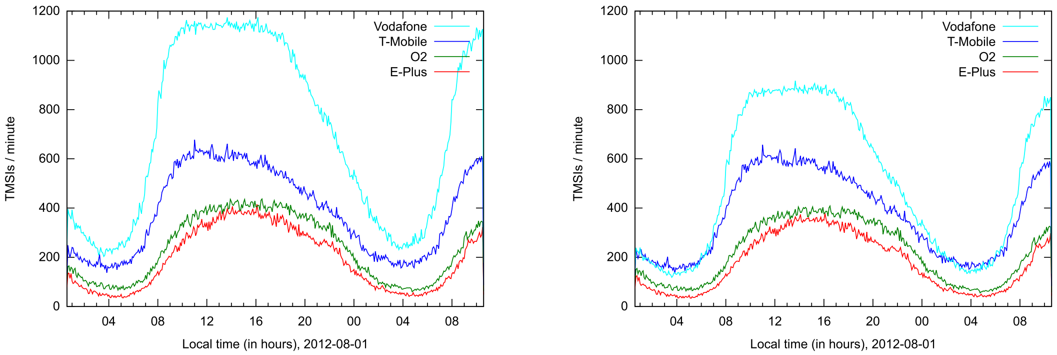
Depending on the network activity, this may or may not be enough to answer all paging requests. Additionally, we need to examine the different paging activities that can be seen in real operator networks. If the paging activity is very large, then the attacker may need to use multiple phones to perform the attack.

Finally, to get an understanding of the impact of such an attack, we need to determine the size of the geographical region covered by a location area.

### 4.1 Location Area Paging Activity

An attack against an entire location area, e.g., in a metropolitan area, requires an adversary to respond to all paging requests in that area. Consequently, the efficiency of a large-scale attack depends on the operator specific paging activities and the allocated resources on the attacker side.

For the purpose of estimating paging activity, we modified our OsmocomBB stack to log all TMSIs in combination with a time stamp of its appearance. Because the paging requests are broadcasted throughout a location area, camping on one operator BTS for that area is sufficient to observe all paging activity for that area on the CCCH/PCH. We recorded the TMSIs in paging requests for all major German operators in a metropolitan area over a time period of 24 hours. The logs were created at exactly the same location, at the same date and



(a) Unfiltered measurement

(b) Filtered paging requests caused by T3113 timer

Figure 5: Number of TMSIs per minute contained in paging requests of four major German operators over 24 hours.

time. We observed that in some cases the network is not paging with the TMSI but with the IMSI. E.g., if the subscriber is marked as attached to the network but cannot be reached using the TMSI, the MSC starts paging using the IMSI. In this case, depending on the operator network configuration, paging may also be performed outside of the location area. However, this type of paging request is the minority and thus ignored in our measurements. Furthermore, assuming that a subscriber is present in the monitored location area, the network very likely already paged using the TMSI in this area. Obviously, it is simple to implement the attack in the case that network pages using IMSIs instead of TMSIs. In fact our code can also handle IMSIs.

Figure 5a summarizes the paging observations. The first observation to be made is that paging activity heavily varies throughout the time of the day. The observed pattern is not random, but rather reflects human activity during typical days. It is also interesting to note that the amount of paging requests heavily differs among the various tested operators. While for example E-Plus at peak times has a rate of roughly 415 TMSIs contained in paging requests per minute, Vodafone has almost 1200 in the same time period.

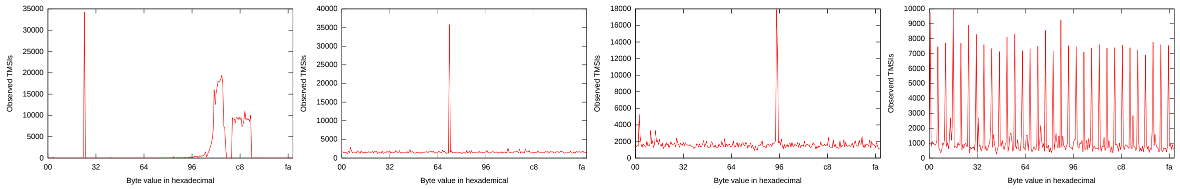
Such differences can be caused for example by the number of active subscribers in the network, or the size of the respective location area. During this measurement, we noticed several reoccurring TMSI patterns. Vodafone is actually always paging each TMSI at least two times. A second paging request is always issued two seconds after the initial paging request. This explains the massive amount of paging requests and we suspect this to be an attempt to improve the overall subscriber availability. Also, our logged data shows that some of these TMSIs are paged at regular intervals. We believe that these requests may partially be directed at M2M devices, e.g., for remote monitoring.

Figure 5b shows a filtered version of Figure 5a. Specifically, we filtered appearances of TMSIs contained in paging requests that we do not need to respond to. 3GPP TS 04.08 [4] specifies a timer, T3113, that is set on transmission of a paging request. If no paging response was received prior to the expiry of this timer, the network reissues the paging request by paging the mobile subscriber again. However, assuming that we are able to observe and respond to all paging requests, this retransmission would not occur during an attack. Therefore, these can be filtered from the result. By analyzing the logged TMSIs and the respective timestamps, we recorded the reappearance of each TMSI that was originally transmitted as part of a paging request. The vast majority of reappearances in time reach a common maximum which we assume is the timer value. A prevalent value seems to be five seconds. It is also reasonable that this is caused by a triggered timer. A normal call setup takes longer than five seconds [30] and short messages are queued at the SMS service center and likely transmitted over the same channel following one paging request.

As a result, the overall activity of relevance in practice is lower than the general amount of TMSIs contained in observed paging requests. The Vodafone measurements can be reduced by almost 22% during peak times and 33% during low activity times. However, due to the limited memory resources of the attacking phones, we cannot take this into account during an active attack.

## 4.2 A Randomized Attack Strategy using TMSIs

The measured data in Section 4.1 suggests that even in location areas with low paging activity an attacker needs more than a single phone to respond to all paging requests. Thus, paging requests need to be distributed across multiple attacking phones. While serial commu-



(a) Byte 3 (b) Byte 2 (c) Byte 1 (d) Byte 0

Figure 6: Statistical distribution of each TMSI byte contained in paging requests for O2. Based on 437734 TMSIs.

nication could be used to coordinate these efforts, it also poses a significant slowdown. Consequently, using serial communication would lower the chance to win the race. We therefore decided to not make use of any actual communication between attacking devices, but to use a probabilistic approach instead.

For this, we analyzed the TMSI values to determine the statistical distribution of each individual TMSI byte as contained in respective paging requests. Namely, to prevent the collection of mobile subscriber identities and thus enable tracking, mobile phones are in most cases identified by their TMSI instead of their IMSI. To provide strong anonymity, a network should therefore sufficiently randomize those short term identities to provide unlinkability. A statistically uniform distribution would ease randomly distributing the paging load across multiple phones. However, an analysis of collected TMSIs made it clear that not all bits of the TMSI are sufficiently random or at least uniformly distributed. This may be, because some parts of the TMSI can be related to, e.g., the time of its allocation [8]. We also observed that certain bytes of the TMSI appear more frequently in specific ARFCNs. Thus, we further analyzed the distribution of each individual of the four TMSI bytes, for all tested operators. We use O2 as an example operator here even though nearly identical patterns can be seen for other carriers.

Figure 6 shows for each possible byte value how often a specific value was found in TMSIs contained in paging requests that we logged. As visible, all byte values are not uniformly distributed. However, 6a, 6b, and 6c show a significantly different pattern from 6d. Not every possible value of the least significant byte (LSB) of the TMSI is encountered with equal frequency on the air interface. For example the value `0xff` is not used at all. Some seem to be more likely than others. Nonetheless, 6d shows that value ranges are close to a uniform distribution. This becomes more plausible in Figure 7, which compares the cumulative distribution function for observed values of the LSB and the uniform distribution. We make use of this characteristic to delegate specific attack phones to dedicated TMSI LSB byte ranges. This way, we can distribute the immense amount of paging between several phones by simply using randomization and thus avoid coordination at all. Outliers for certain value ranges could be compensated by adding more phones to the specific range. To prevent recompilation of our OsmocomBB based firmware for distinct value range, we introduced a mechanism to configure the range at runtime. This mechanism is similar to the TMSI setting described in Section 3.4 and is based on a HDLC message over serial.

A similar distribution could be achieved by hashing TMSI values and assigning individual phones to specific hash prefixes. However for simplicity and to reduce the response time as much as possible we decided not to do this.

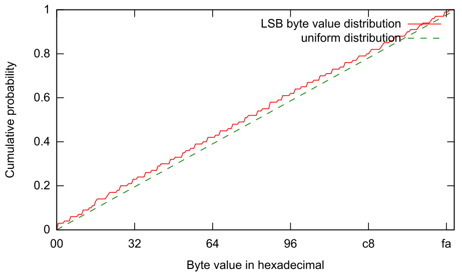


Figure 7: Cumulative distribution function for Byte 0 (LSB) of TMSIs contained in paging requests observed for O2.

## 4.3 Mapping Location Areas

When performing a large-scale attack against a geographic region, we have to determine the size covered by the location area. Specifically, this knowledge enables an adversary to precisely plan the affected zone of such an attack. An attacker carefully selects the target location areas for specific regions and operators.

Location areas are not organized to cover an equally large area. As pointed out in Section 4.1, this impacts the paging activity that can be observed in a specific location area. Their size differs among operators and specifics of the covered environment. In fact, because of its impact on mobility management, location area planning is an important aspect for mobile network operators. Its size manifests a trade-off between subscriber-induced and network-induced performance degradation. Small location areas can cause a significant signaling overhead in the core network due to frequent location updates. It has already been demonstrated, that this can lead to denial of service like conditions [36]. A large location area causes additional load due to the paging overhead.

The Location Area Code (LAC), which is part of the Location Area Identifier (LAI), is broadcasted by each BTS in regular intervals on the BCCH via a *System Information Type 3* message. To map location areas, we use a slightly modified version of the cell_log application from the OsmocomBB tool-chain. cell_log scans all ARFCNs in the assigned GSM frequency spectrum for a carrier signal. It then attempts to sync to these frequencies and logs decoded system information messages as broadcasted on the BCCH. In combination with off-the-shelf GPS receivers, we determine the geographic location of the observed LAC.

By slowly driving through the city in a car, we collected a number of waypoints and the respective GSM cells in sight. To minimize the loss due to driving speed, the scan process was performed simultaneously on eight OsmocomBB devices. In order to estimate the surface covered by a location area, we calculated the convex hull of points within the same LAC. The size of location areas in a metropolitan area such as Berlin varies from 100km$^2$ to 500km$^2$. From our study, the average location area in Berlin covers around 200km$^2$. Data from OpenCellID and Crowdflow [29, 31] indicate that outside of the city center location areas exist that cover over 1000km$^2$. Figure 8 shows location areas that we mapped for one of the four major operators in Berlin.

Most location areas partially overlap with geographic regions that are part of a different area. These results provide a rough insight on dimensions of location areas in a metropolitan area. It also shows that a large-scale denial of service attack based on the paging procedure has a significant impact to a large number of subscribers.

## 4.4 Amplification of the Paging Response Attack

The attack procedure as introduced in Section 3 does prevent MT services from being delivered to a subscriber. However, it does not provide a persistent way to cause denial of service conditions. Access to mobile services is denied as long as an adversary is running the attack. Accordingly, calls reissued by subscribers to reach a person, have to be attacked again, which may further raise the paging load. To prevent this, we make use of another attack that has been publicized before. Munaut discovered that the *IMSI DETACH* message is not authenticated in GSM and 3G networks [34]. As a result, an attacker can easily craft detach messages on behalf of a victim. This message acts as an indication to the network that a subscriber is no longer marked as available for the carrier. As a result, the network marks the mobile station as detached and will no longer page the subscriber until it reassociates with the network. Consequently, this stops the network from delivering MT services. During normal operation, this message is generated by the phone and sent to the network, e.g., when it is being switched off.

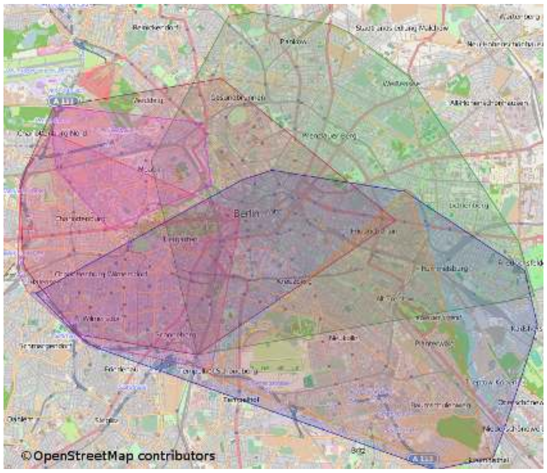


Figure 8: Location Areas of Vodafone Germany in Berlin.

The mobile identity contained in the detach indication message is not limited to IMSIs, but can also contain TMSIs. By combining the paging response attack with the IMSI detach attack, it is therefore possible to amplify its effect. After each paging response, our OsmocomBB implementation reuses the collected mobile identity to send a detach message. Accordingly, our attack ensures that an initial call to a subscriber will be terminated and that reissued services such as calls will not cause paging activity again. Thus, by doing so, we effectively reduce the paging load over time.

## 4.5 Large-scale Attack Feasibility

We continue to evaluate the feasibility of a large-scale attack using multiple phones against commercially deployed networks. The transmission of a large number of RACH bursts and SDCCH channel allocations may be limited due to radio resource bottlenecks. We therefore verify, whether or not a single cell provides enough resources, or an attack needs to be conducted in a distributed fashion.

**Prerequisites.** In the following, we denote the TMSI paging request activity as $r_{\text{request}}$ and the number of required phones to handle this respectively as $n_{\text{phones}}$. As discussed in Section 4.2, the TMSI LSB value range is used to equally distribute the paging load across multiple attacking phones. Therefore, assigned phones need to wait for a range match. On average this requires $t_{\text{matching}} = r_{\text{request}}/n_{\text{phones}}$ seconds. On a match, the phone sends a channel request on the RACH and a paging response on an SDCCH in $t_{\text{response}}$ seconds. It finally synchronizes back to the CCCH in $t_{\text{sync}}$ seconds to be prepared for the next run. The required time for an attack is therefore $t_{\text{attack}} = t_{\text{matching}} + t_{\text{response}} + t_{\text{sync}}$ seconds. Thus, for a successful attack, the minimum number of phones an adversary requires is $n_{\text{phones}} \geq r_{\text{request}} \cdot t_{\text{attack}}$.

**RACH Resource Constraints.** The available resources provided by a single cell depend on its configuration. The GSM specifications defines a number of valid channel configurations [7]. Thus, an adversary is limited by the number of available RACH slots and the number of SDCCHs that a cell provides. In practice cells in metropolitan areas use the *BCCH+CCCH* or *FCCH+SCH+CCCH+BCCH* channel configurations on the first time slot. These are not combined with DCCHs and therefore allow all 51 bursts on the uplink of a 235.4 ms 51-multiframe to be used to transmit channel requests on the RACH. Because the RACH is a shared medium, collisions with requests of other subscribers may occur. According to Traynor et al. [36], the maximum resulting throughput is 37%. As a result, an attacker can transmit up to $r_{\text{RACH}} = 51/0.2354 \cdot 0.37 \approx 80$ channel requests per second in a single cell. Consequently, given that $n_{\text{phones}} \leq n_{\text{RACH}} = r_{\text{RACH}} \cdot t_{\text{attack}}$ is true, a single cell can fulfill the channel request requirements.

**SDCCH Resource Constraints.** Following the channel allocation, the adversary phone uses an SDCCH to send the paging response. Analogical to the RACH, SDCCHs in medium- or large-sized cells in a metropolitan area are provided on a separate time slot. A typical *SDCCH/8+SACCH/8* channel configuration comprises of 8 SDCCHs per 51-multiframe, in theory offering: $r_{\text{SDCCH}} = 8/0.2354 = 34\,\text{SDCCH/second}$. Clearly this may make the signaling channel the major bottleneck for this attack. Accordingly, the occupation time of these channels needs to be taken into account. For example according to Traynor et al., a rough estimation of the occupation time of the channel for an Insert Call Forwarding operation is 2.7 seconds [36]. Compared to this, our attack occupies the channel for a very short duration, as shown in Section 3.5.

Similar to the RACH requirements, the maximum number of attacking phones per cell is therefore bounded by $n_{\text{phones}} \leq n_{\text{SDCCH}} = r_{\text{SDCCH}} \cdot t_{\text{attack}}$.

**Example Computation.** The following is an example, based on the peak values from our measurements gathered for the E-Plus network and as reflected in Table 2. Based on the previous equations, at least $n_{\text{phones}} \approx 10.820$ phones are required to attack a typical location of E-Plus. Given the costs of the Motorola devices, this is a reasonably small amount. Each paging response attack lasts $t_{\text{attack}} \approx 1.564$ seconds. This allows up to $n_{\text{RACH}} \approx 125$ phones without a saturation of the RACH. For the SDCCH, the above formula yields to a maximum of $n_{\text{SDCCH}} \approx 53$ phones. It is also important to note that the number of phones is proportional to the impact. This means that half of the attacking phones would still be able disrupt service for half of the subscribers of a location area.

Table 2: Example resource requirements for E-Plus.

| Variable | Value | Reference |
|---|---|---|
| $r_{request}$ | 415 paging/min | Section 4.1 |
| $t_{response}$ | 180 ms | Section 3.5 |
| $t_{sync}$ | 745 ms | Section 4 |

A single cell therefore provides enough resources to attack a complete location area of a considerably small operator. In practice these resources are shared with legit MO and MT traffic. The exact traffic patterns and the number of cells per location is unknown. Furthermore, a combination with the IMSI detach attack prevents phones that reside in the location area to generate further MT activity. As we cannot estimate these activities, we do not include this in our calculation. Nevertheless, the results indicate the required resources for a large-attack do not extensively exhaust the resources provided by a cell. Additionally, there is no technical limitation of distributing attacking phones across small number of different cells.

## 5 Countermeasures

In this section we present two countermeasures against the attacks we developed. Specifically, we propose different approaches to resolve both problems. A solution is required to not only fix the denial of service issue, but at the same time the MT service hijacking. Unlike the second prevention strategy, the first solution solves both issues at once, but requires a protocol change.

For the first solution, we propose a change to the paging protocol procedure [4]. To perform authentication, the network is sending a 128 bit random challenge (RAND) to the subscriber. Based on the secret key $K_i$ that is only stored on the SIM card or in the authentication center of the network, the subscriber computes a 32 bit response value using the A3 algorithm. The so-called Signed Response (SRES) value is sent back to the network. In the same fashion, the operator network computes SRES based on $K_i$ as stored in the authentication center. If both SRES values match, the subscriber successfully authenticated itself to the network. However, as mandated in the GSM specification, the authentication is performed after the paging response is processed. The same principle applies to UMTS [12]. Therefore, the paging response itself is not authenticated. By adapting the protocol to include the RAND value in the paging request and SRES in the paging response, this can be changed. This implies that all of the paging responses are authenticated, which eliminates session hijacking. At the same time a paging response that includes authentication information can be used by the network to validate the response before changing the state to not expect further paging responses. Thus, also solving the denial of service attack. It is important to note that this requires a fresh RAND for every authentication to prevent replay attacks. This is similar to the protocol change proposed by Arapinis et al. [16], which encrypts the paging request using a shared session key called *unlikability key*. While they use this key to prevent tracking of subscribers via IMSI paging, the same modification also prevents our described attacks. Unfortunately, partly due to the difficulty of updating devices in the field, the industry is reluctant to apply new protocol changes to commercially deployed networks.

The second solution involves no protocol change, but has to dismantle each problem individually. MT session hijacking issue can be addressed, by enforcing authentication for each service request. This would also overcome MO session hijacking. In order to eliminate the denial of service attack, the MSC/VLR state machine needs to be changed. Specifically, the MSC/VLR has to be able to map all incoming paging responses to the correct service as long as no fully authenticated session exists. Accordingly, this circumvents the denial of service attack.

## 6 Related Work

In the last years, various attacks against cellular networks and their protocol stacks have been published. We separate related work into two parts. First, attacks that allow an adversary to impersonate a victim. Second, denial of service attacks in mobile networks that result in customers not being able to receive MT services.

**Impersonation.** In [28] Nohl and Melette demonstrated that it is possible to impersonate a subscriber for mobile originated services. By first sniffing a transaction over-the-air, cracking the session key $K_c$, and knowing a victims TMSI, they were able to place a phone call on behalf of a victim. The authors of [24] used a femtocell device under their control in order to impersonate a subscriber that is currently booked into the femtocell. By relaying authentication challenges to a victim, they were able to send SMS messages on behalf of the subscriber. Our work in this paper is different, as we do not attack MO services, but MT services. Thus, in our research, the considered victim is, e.g., the called party and not the caller. Contrary to attacking MO services, attacking MT services is time critical.

**Denial of Service.** We consider relevant types of denial of service attacks in mobile networks that affect MT services for subscribers. We determined three types of denial of service attacks that fulfill this requirement: attacks directly targeting the victim phone, attacks focusing on the network, and attacks affecting subscribers, but without direct communication.

The first type comprises DoS attacks that target mobile devices directly, most notably phones. These issues are usually baseband/phone specific and caused by implementation flaws. Several vulnerabilities have been discovered in mobile phones that can lead to code execution and denial of service conditions [33, 46]. Particularly, the Curse-of-Silence flaw enabled an adversary to disable the MT SMS functionality of specific Nokia devices [44]. In [38] Racic et al. demonstrate that it is possible to stealthily exhaust mobile phone batteries by repeatedly sending crafted MMS messages to a victim. Consequently, the phone battery will drain very fast, eventually the phone will switch off, and MT services can no longer be delivered to a victim. Our attack is inherently different from these kinds of attacks, because it is independent from the target device type and does not interact with the victim directly at all.

The second category consists of attacks that target the operator itself, and as a consequence also impact MT services for subscribers. These types are caused by design flaws. Spaar showed in [43] that it is feasible to exhaust channel resources of a base station by continuously requesting new channels on the RACH. Unlike our attack, this attack is limited to a single BTS and does not affect subscribers served by a different cell. Therefore, to attack a metropolitan area, an attacker needs to communicate with and attack every BTS in that area. Enck et al. [21] showed that it is practical to deny voice or SMS services within a specific geographical area, by sending a large number of short messages to subscribers in that area. Serror et al. [42] exhibit that similar conditions can be achieved in CDMA2000 networks by causing a significant paging load and delay of paging messages via Internet originating packets to phones. A comparable resource consumption attack for 3G/WiMax has been demonstrated by Lee et al. in [32]. As Traynor et al. outline [36], it is also possible to degrade the performance of large networks by utilizing a phone botnet and, e.g., repeatedly adding and deleting call forwarding settings. All of these attacks exhaust network resources mostly due to generated signaling load. As a result, services can no longer be reliably offered to mobile subscribers, effectively causing denial of service conditions. This includes MT and MO services. We exploit a race condition in the MT paging procedure and do not attack the core network itself. Our attack does not intend to generate excessive signaling traffic in the network. As a result, it is not prevented by proposed mitigations for these kind of issues from previous research.

Our attack fits into the last of the three types of attacks that result in DoS for MT services. Most network attacks aim to abuse generated signaling to decrease the overall performance of the operator network. Attacks against mobile devices merely use the network as a bearer to deliver a specific payload to the phone. The third category is stipulated by attacks that target the mobile device itself, but do not send any payload to it. The aforementioned IMSI detach attack discovered by Munaut [34] can effectively cause that a service such as a call, will not result in paging requests by the network anymore. As described in section 4.4, this design flaw even supports our attack. Contrary to this vulnerability, the paging response attack allows us to precisely control when and where a victim can be reached or not. After sending a detach indication, an attacker cannot control anymore for how long this state is kept.

Our approach can be used either to hijack a session or to perform a denial of service attacks. We do attack mobile stations but neither by exhausting network resources, nor by directly communicating to the target device. We can target specific geographical areas, specific subscribers or a group of subscribers without the need to build a hit list of phone numbers residing in that area. Depending on the target, the attack can be either distributed or performed from a single phone. Additionally, the involved costs for this attack are as cheap as acquiring the required number of Motorola C1XX phones.

## 7 Conclusion

The trust in the security of cellular networks and specifically the widely used GSM standard has been shattered several times. Yet, attacks against mobile terminated services are a minority. The undisturbed operation of telecommunication networks is traditionally based on trust. The inherent trust that each subscriber and participant in communication plays by the rules. Nonetheless, due to several available and modifiable software and hardware projects for telecommunication, this trust relationship has to be considered broken. In this paper we showed how to exploit the trust in paging procedures on a broadcast medium. We demonstrated that it is possible to leverage a race condition in the paging protocol to a novel denial of service attack and the possibility to hijack mobile terminated services in GSM. Moreover, we showed that this attack can not only disturb communication for single subscribers, but can also greatly affect telephony in a larger geographical region formed by location areas. A motivated attacker can interrupt communication on a large scale by merely utilizing a set of inexpensive consumer devices that are available on the market. This is considerably more efficient compared to traditional radio jamming due to the broad frequency range of mobile carrier networks and the size of location areas. In order to mitigate these attacks, we propose two different countermeasures of which one does not require a protocol change. We strongly encourage future standards to consider threats caused by active attackers that tamper with user equipment and protocol stacks.

## 8 Acknowledgement

The authors would like to thank Dieter Spaar, Harald Welte, Holger Freyther, Benjamin Michéle, Alex Dent, and Dmitry Nedospasov for technical discussions and their help in reviewing this paper.

## References

[1] Routo Messaging. http://www.routomessaging.com.

[2] Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth. http://www.gartner.com/it/page.jsp?id=1924314, February 2012.

[3] 3GPP. Digital cellular telecommunications system (Phase 2+); Network architecture (GSM 03.02 version 7.1.0 Release 1998). Tech. rep., 3rd Generation Partnership Project, 2000. 3GPP TS 03.02 V7.1.0.

[4] 3GPP. Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 specification (3GPP TS 04.08 version 7.9.1 Release 1998). Tech. rep., 3rd Generation Partnership Project, 2001. 3GPP TS 04.08 V7.9.1.

[5] 3GPP. Digital cellular telecommunications system (Phase 2+); Multiplexing and multiple access on the radio path (3GPP TS 05.02 version 8.9.0 Release 1999). Tech. rep., 3rd Generation Partnership Project, 2001. 3GPP TS 05.02 V8.9.0.

[6] 3GPP. Digital cellular telecommunications system (Phase 2+); Base Station System - Mobile Services Switching Centre (BSS-MSC) Interface - Interface Principles (3GPP TS 08.02 version 8.0.1 Release 1999). Tech. rep., 3rd Generation Partnership Project, 2002. 3GPP TS 08.02 V8.0.1.

[7] 3GPP. Digital cellular telecommunications system (Phase 2+); Mobile Station - Base Station System (MS - BSS) Interface Channel Structures and Access Capabilities (3GPP TS 04.03 version 8.0.2 Release 1999). Tech. rep., 3rd Generation Partnership Project, 2002. 3GPP TS 04.03 V8.0.2.

[8] 3GPP. Digital cellular telecommunications system (Phase 2+); Numbering, addressing and identification (3GPP TS 03.03 version 7.8.0 Release 1998). Tech. rep., 3rd Generation Partnership Project, 2003. 3GPP TS 03.03 V7.8.0.

[9] 3GPP. Digital cellular telecommunications system (Phase 2+); Security-related network functions (3GPP TS 03.20 version 8.6.0 Release 1999). Tech. rep., 3rd Generation Partnership Project, 2008. 3GPP TS 03.20 V8.6.0.

[10] 3GPP. Digital cellular telecommunications system (Phase 2+); Radio transmission and reception (3GPP TS 45.005 version 9.1.0 Release 9). Tech. rep., 3rd Generation Partnership Project, 2010. 3GPP TS 45.005 V9.1.0.

[11] 3GPP. Universal Mobile Telecommunications System (UMTS);Physical channels and mapping of transport channels onto physical channels (FDD)(3GPP TS 25.211 version 9.2.0 Release 9). Tech. rep., 3rd Generation Partnership Project, 2010. 3GPP TS 25.211 9.2.0.

[12] 3GPP. Universal Mobile Telecommunications System (UMTS); LTE;3G security; Security architecture(3GPP TS 33.102 version 9.4.0 Release 9). Tech. rep., 3rd Generation Partnership Project, 2011. 3GPP TS 33.102 V9.4.0.

[13] 3GPP. Digital cellular telecommunications system (Phase 2+); Mobile Switching Centre - Base Station system (MSC-BSS) interface; Layer 3 specification (3GPP TS 48.008 version 9.8.0 Release 9). Tech. rep., 3rd Generation Partnership Project, 2012. 3GPP TS 48.008 V9.8.0.

[14] 3GPP. LTE;Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode(3GPP TS 36.304 version 9.9.0 Release 9). Tech. rep., 3rd Generation Partnership Project, 2012. 3GPP TS 36.304 V9.9.0.

[15] 3GPP. Universal Mobile Telecommunications System (UMTS);User Equipment (UE) procedures in idle mode and procedures for cell reselection in connected mode(3GPP TS 25.304 version 9.8.0 Release 9). Tech. rep., 3rd Generation Partnership Project, 2012. 3GPP TS 25.304 V9.8.0.

[16] Arapinis, M., Mancini, L., Ritter, E., Ryan, M., Golde, N., Redon, K., and Borgaonkar, R. New Privacy Issues in Mobile Telephony: Fix and Verification. In *Proceedings of the 19th ACM Conference on Computer and Communications Security* (October 2012).

[17] Barkan, E., Biham, E., and Keller, N. Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. *J. Cryptol. 21*, 3 (Mar. 2008), 392–429.

[18] Biryukov, A., Shamir, A., and Wagner, D. Real Time Cryptanalysis of A5/1 on a PC. In *Proceedings of the 7th International Workshop on Fast Software Encryption* (London, UK, UK, 2001), FSE '00, Springer-Verlag, pp. 1–18.

[19] Boswarthick, D., Elloumi, O., and Hersent, O. *M2M Communications: A Systems Approach*. Wiley, March 2012.

[20] D. Burgess et al. OpenBTS. `http://openbts.org`.

[21] Enck, W., Traynor, P., McDaniel, P., and La Porta, T. Exploiting open functionality in SMS-capable cellular networks. In *Proceedings of the 12th ACM conference on Computer and communications security* (New York, NY, USA, 2005), CCS '05, ACM, pp. 393–404.

[22] Ettus. USRP. `http://www.ettus.com/products`, 2009.

[23] Frank A. Stevenson. [A51] The call of Kraken. `http://web.archive.org/web/20100812204319/http://lists.lists.reflextor.com/pipermail/a51/2010-July/000683.html`, July 2010.

[24] Golde, N., Redon, K., and Borgaonkar, R. Weaponizing Femtocells: The Effect of Rogue Devices on Mobile Telecommunications. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium* (Feb. 2012).

[25] H. Welte. OpenBSC. `http://openbsc.osmocom.org`.

[26] Infosecurity Magazine. Indian company hacks GSM and usurps IMSI. `http://www.infosecurity-magazine.com/view/24680/indian-company-hacks-gsm-and-usurps-imsi/`, March 2012.

[27] ip.access Ltd. nanoBTS 1800. `http://www.ipaccess.com/picocells/nanoBTS_picocells.php`.

[28] Karsten Nohl and Luca Melette. Defending mobile phones. `http://events.ccc.de/congress/2011/Fahrplan/events/4736.en.html`, December 2011.

[29] Krell, M. Crowdflow. `http://crowdflow.net`.

[30] Kune, D. F., Koelndorfer, J., Hopper, N., and Kim, Y. Location leaks over the GSM air interface. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium* (Feb. 2012).

[31] Landspurg, T. OpenCellID. `http://opencellid.org`.

[32] Lee, P. P. C., Bu, T., and Woo, T. On the detection of signaling DoS attacks on 3G/WiMax wireless networks. *Comput. Netw. 53*, 15 (2009), 2601–2616.

[33] Mulliner, C., Golde, N., and Seifert, J.-P. SMS of Death: From Analyzing to Attacking Mobile Phones on a Large Scale. In *Proceedings of the 20th USENIX Security Symposium* (San Francisco, CA, USA, August 2011).

[34] Munaut, S. IMSI DETACH DoS. `http://security.osmocom.org/trac/ticket/2`, May 2010.

[35] Nokia Siements Networks. Nokia Siemens Networks promotes GSM for Machine to Machine applications. `http://www.nokiasiemensnetworks.com/news-events/press-room/press-releases/nokia-siemens-networks-promotes-gsm-for-machine-to-machine-applications`.

[36] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, T. La Porta, P. McDaniel. On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core. In *ACM Conference on Computer and Communications Security (CCS)* (November 2009).

[37] PurpleLabs. Tsm30 firmware. `http://web.archive.org/web/20090325133430/http://sourceforge.net/projects/plabs`, November 2004.

[38] Racic, R., Ma, D., and Chen, H. Exploiting MMS Vulnerabilities to Stealthily Exhaust Mobile Phone's Battery. In *Securecomm and Workshops, 2006* (28 2006-sept. 1 2006), pp. 1 –10.

[39] Security Research Labs. A5/1 decryption project. `http://opensource.srlabs.de/projects/a51-decrypt`.

[40] Security Research Labs. Decrypting GSM phone calls. `https://srlabs.de/decrypting_gsm/`.

[41] Security Research Labs. GSM security map. `http://www.gsmmap.org`.

[42] Serror, J., Zang, H., and Bolot, J. C. Impact of paging channel overloads or attacks on a cellular network. In *Proceedings of the 5th ACM workshop on Wireless security* (New York, NY, USA, 2006), WiSe '06, ACM, pp. 75–84.

[43] Spaar, D. RACH flood DoS. `http://security.osmocom.org/trac/ticket/1`, November 2009.

[44] T. Engel. Remote SMS/MMS Denial of Service - Curse Of Silence. `http://berlin.ccc.de/~tobias/cursesms.txt`, December 2008.

[45] Various contributors. Osmocom project. `http://osmocom.org`.

[46] Weinmann, R.-P. Baseband Attacks: Remote Exploitation of Memory Corruptions in Cellular Protocol Stacks. In *Proceedings of the 21st USENIX Workshop on Offensive Technologies* (Bellevue, WA, USA, August 2012).

[47] Welte, H., Munaut, S., Eversberg, A., and other contributors. OsmocomBB. `http://bb.osmocom.org`.