Real-Time Passive Capturing of the GSM Radio

Islam Alyafawi
University of Bern
Bern, Switzerland
Email: alyafawi@iam.unibe.ch

Desislava C. Dimitrova
University of Bern
Bern, Switzerland
Email: dimitrova@iam.unibe.ch

Torsten Braun
University of Bern
Bern, Switzerland
Email: braun@iam.unibe.ch

***Abstract*—This paper addresses the problem of service development based on GSM handset signaling. The aim is to achieve this goal without the participation of the users, which requires the use of a passive GSM receiver on the uplink. Since no tool for GSM uplink capturing was available, we developed a new method that can synchronize to multiple mobile devices by simply overhearing traffic between them and the network. Our work includes the implementation of modules for signal recovery, message reconstruction and parsing. The method has been validated against a benchmark solution on GSM downlink and independently evaluated on uplink channels. Initial evaluations show up to 99% success rate in message decoding, which is a very promising result. Moreover, we conducted measurements that reveal insights on the impact of signal power on the capturing performance and investigate possible reactive measures.**

I. Introduction

In recent years, wireless devices are getting more powerful and pervasive. Besides that, there is an increasing number of developed services, such as location based services (LBSs), targeting these devices. Services run on top of supporting systems. A system is described as active when the service deployment is provisioned at the mobile terminal [8], [9] or at the network side [10], [11]. In both cases, cooperation is required either from the end user or the network operator. A passive system on the contrary does not require participation of the communicating parties but relies on overhearing radio signals and their subsequent processing.

Current wireless devices often support more than one radio technology, e.g., WiFi, Bluetooth and the Global System for Mobile Communications (GSM). The wide availability of GSM networks encourages research on the use of GSM as a common radio technology for service development. In addition, GSM signals appear more stable over time in comparison to WiFi or Bluetooth signals [12], which is a crucial factor in the quality of the service. In this paper we also opted for the use of GSM.

The design of a passive-based localization service has several challenges that are related to the nature of the wireless medium and the GSM standards. First, how can we capture GSM radio signals, convert them to messages and parse the message content? Second, can we identify the signal source in order to provide the correct service? Facing these challenges requires an uplink receiver that captures, processes and analyzes GSM radio signals generated by the mobile devices.

This paper offers a receiver of GSM uplink signals. The developed GSM receiver overcomes the challenges of (i) synchronization with the end users in time and frequency, (ii) signal power recovery and (iii) message parsing. Besides identifying mobile devices, the receiver facilitates feedback on the Received Signal Strength (RSS) as an important measure in many passive applications, such as localization. To our knowledge, until now, there is no passive tool that can offer comprehensive capturing and interpretation of GSM uplink signals. Although OpenBTS [5] implements the GSM radio interface for uplink, it relies on communication with the users. Our system presents the first effort of a GSM receiver development that can be used for the purpose of passive location services.

A series of experiments were conducted in real GSM networks for performance evaluation of the uplink receiver. The receiver shows a reliable performance of signal recovering with a success rate up to 99%. Such a system remains invisible to the target devices and is hence attractive for third parties, which wish to avoid dependency on network operators to provide their services. We are aware that privacy questions may raise and detailed investigations on data anonymization will be interesting. In our current solution, as a first step, we work with objective identifiers, which we (as non operator) cannot relate to an identity.

In the following, Section II summarizes the challenges of signal capturing for GSM-based services, while Section III introduces existing tools. Section IV presents the developed passive uplink receiver, which is evaluated in Section V. Section VI concludes the work.

II. Passive receiver Requirements

A passive receiver relies on the concept of signal overhearing, in which the communication between two radio devices is overheard by a third device for the purpose of a specific application. The system operation is illustrated in Figure 1. A mobile station (MS) communicates with a base station (BS) and the traffic is overheard by a number of passive nodes, termed anchor nodes (ANs). ANs do not have any information about the MS location or transmission time and frequency. Ideally, BSs cover geographical areas with hexagonal shapes. MSs located within a BS's coverage area communicates only with that BS. A passive receiver system has to deal with several challenges before becoming a commonly adopted solution.

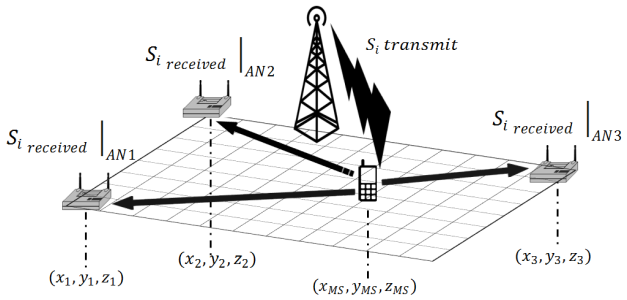

Fig. 1. General design of a passive receiver system

Radio signal acquisition is among the initial requirements and is the focus of this paper.

A. GSM Signal Capturing

One of the challenges of a GSM-based passive receiver is to capture the radio signal of a specific user. In GSM, information is divided into blocks referred to as bursts, which fit into a single time slot (TS) of the GSM time frame [13]. Therefore, signal capturing requires frequency and time synchronization between transmitter and receiver. In order to support the MS in this task, the BS periodically transmits two burst types:

Frequency correction Burst (FB) indicates the carrier frequency of the broadcasting physical channel of the cell. It is used by the mobile station for initial frequency synchronization and subsequent re-tuning.

Synchronization Burst (SB) is used to support time synchronization between mobile station and base station. The SB contains a globally unique Training Sequence Code (TSC) of 64 bits, specifically designed for time alignment.

Time alignment is additionally supported by a Timing Advance (TA) parameter. Since users, served by the same cell, can be at different distances from the base station, their transmission will not perfectly aligned with the GSM time frame due to propagation delays. To compensate for the delays, the BS sends specific TA values to correct the mobile stations transmit start.

Normal Bursts (NBs) are used to carry control and user traffic exchanged between the MSs and their serving base station. A NB contains two information blocks separated by a 26-bit midamble, which carries one of eight predefined TSCs; see Figure 2. Since this TSC is different from the training sequence in a synchronization burst, we introduce the notations TSC(NB) and TSC(SB), respectively. TSC(NB) is typically used for fine-grained time correction and can be obtained from the BSIC transmitted by the serving BS [3].

An active system, running at either the mobile station or the GSM network, can use the availability of FB and SB to reconstruct the carried messages. In a passive system, however, the MS does not cooperate and offer any information on frequency and time synchronization to recover uplink messages. Different TA values are an additional difficulty. To tackle the challenge of synchronization to the mobile user, we propose in Section IV an approach relying on processing of Normal Bursts to tune to individual users.

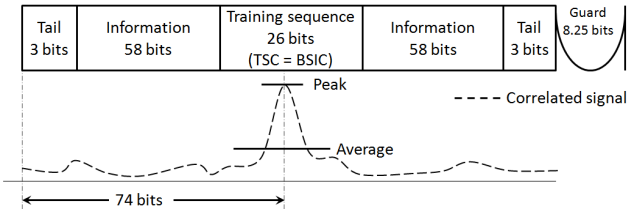

Fig. 2. Normal burst structure

B. GSM user identification

In GSM, a MS can be identified by its International Mobile Subscriber Identity (IMSI) or Temporary Mobile Subscriber Identity (TMSI). Therefore, any passive receiver system, after capturing GSM radio signals, should reconstruct the carried message to detect the presence of one of the two identifiers. In a preliminary investigation, we identified few procedures such as MS registration, periodic or roaming location update, or service request that contain useful messages including IMSI or TMSI. In particular, the messages Paging Response and Location Update Request passed in non encrypted mode on the air interface are of special interest as they carry a MS identity. Experimental analysis of these messages is presented in Section V.

The first stage in message reconstruction is converting an analog signal to a digital bit sequence, carried out at the physical layer as illustrated in Section III. Next, since GSM uses time multiplexing with a complex multiframe structure, we need to recombine bursts into messages. A single message on the data link layer (184 bits) maps to 456 bits on the physical layer, which fit into four Normal Bursts located in four consecutive TDMA frames [13]. Failing to obtain the correct order of NBs or missing one of them would prevent the message reconstruction. Section IV describes our solution to overcome this problem. Finally, the combined data should be interpreted according to the GSM standard. For this purpose, a message parser should be implemented. While modules for signal processing of the GSM physical layer are available, modules for message reconstruction and parsing in the uplink are more difficult to find. In fact, at the time of our research no such tools were available.

In particular, we found the following challenges for the message reconstruction process:

- Low signal-to-noise-and-interference ratio (SNIR) - the total amount of signal received power at the anchor terminal P_{Rx} is influenced by power control in GSM uplink as well as transmission path loss, channel noise N, and interference I from i_n co-channel cells. Successful message decoding requires a SNIR above a certain threshold, formally expressed by:

$$\frac{P_{Rx}}{N + \sum_{i=1}^{i_n} I} \geq \text{SNIR} \mid_{\text{decodable message}}$$

The SNIR issue is further investigated in Section V.

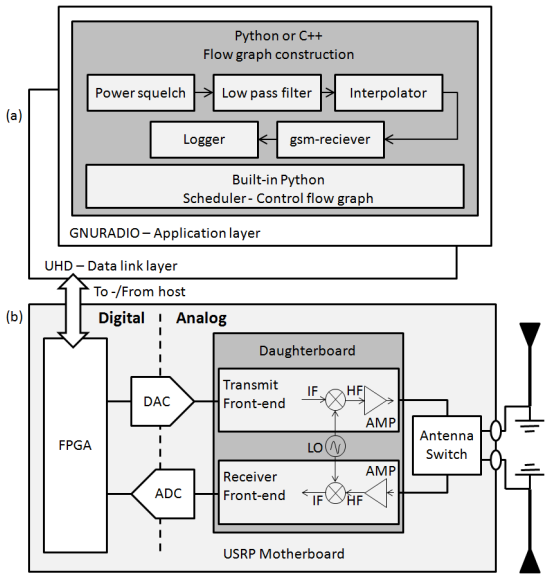

Fig. 3. Simplified block diagram of USRP-based SDR system.

- Time and frequency misalignment - imprecise synchronization of the anchor node and the mobile device leads to erroneous reconstruction of messages, rendering them either unreadable or falsely parsed. The issues are discussed in detail in Section IV.
- Message encryption - encryption effectively hides user data that is potentially useful for identification. The current work is limited to unencrypted messages.

III. GSM SIGNAL CAPTURING TOOLS

Software Defined Radio (SDR) offers an excellent opportunity to develop flexible systems for signal capturing and analysis. Basically, an SDR system consists of radio front end (implemented in hardware) and signal processing modules (implemented in software).

The Universal Software Radio Peripheral (USRP) by Ettus Research [4] is one of the most popular, inexpensive hardware platforms to host an SDR system. The USRP hardware is controlled via an open source USRP Hardware Driver (UHD). UHD translates instructions between the hardware components of the Field-Programmable Gate Array (FPGA) and the signal processing software, as illustrated in Figure 3-b.

Concerning SDR toolkits, the open source GNU Radio framework offers most general functionality for signal acquisition. C++ is used for the various signal processing blocks, while GNU Radio applications are primarily written in Python as shown in Figure 3-a. Other modules can run on top of GNU Radio for customized signal processing and protocol-specific data parsing. The two projects, which are most relevant to us, are Airprobe and OpenBTS.

Airprobe. Airprobe is an open-source project aiming to intercept and parse signals in the GSM downlink using an USRP device. It contains (1) USRP source, Low Pass Filter (LPF) and Interpolator modules (taken over from GNU Radio) and (2) gsm-receiver and gsm-decoder. The *gsm-receiver* module contains the signal processing chain, which demodulates, deinterleaves and decodes the captured GSM signal. The resulting bits are parsed to meaningful parameters inside the *gsm-decoder* module. Airprobe can help to gain knowledge about the serving cell (e.g., the serving BSIC, channel frequency-assignment, or general statistics on user identities) but cannot be used to analyze any uplink traffic. Moreover, it is a deprecated project.

OpenBTS. OpenBTS is another open source project which emulates GSM functionality. It implements the GSM air interface stack of the base station up to layer 3. Since an USRP attached to OpenBTS acts as a base station, mobile users can connect to it following the same standard procedure like in a normal GSM network. Therefore, uplink signaling is ensured to be synchronized based on the frequency and multiframe timing indicated by OpenBTS. OpenBTS has two main drawbacks concerning our target - (1) operating in the GSM radio spectrum requires obtaining licences, and (2) OpenBTS communicates over the air with the target mobile devices. Both do not meet the requirements of a passive system for signal capturing.

IV. PASSIVE GSM RECEIVER

The first step towards GSM signal reconstruction is system synchronization. Frequency synchronization is easily done but time synchronization is more challenging. This is best illustrated by an example. As mentioned in Section II-A, each mobile device has an individual TA value to compensate for its distinct delay in reaching the base station. Let there be two devices MS1 and MS2 at distances $d1$ and $d2$ from the base station. There will be two corresponding TA1 and TA2 values. Unless co-located with the base station, a passive receiver will be located at the different $d1'$ and $d2'$ distances from MS1 and MS2, respectively. As a result, transmissions from the two devices are asynchronous to the passive uplink receiver. Hence, we cannot apply periodic time frame for reception and messages from two MSs may overlap at the receiver terminal.

A. Synchronization Challenges

An ideal solution for the time synchronization challenge outlined in Section III is to intercept downlink and uplink signals simultaneously. The allocated TS and TA transmitted on a downlink channel can feedback to the uplink receiver for time synchronization. The implementation of such a system through commonly used SDR hardware, e.g., USRP, faces several hardware limitations in our experience.

- *Radio Front End* The currently available USRP hardware does not allow two front end receivers to be tuned individually for capturing GSM uplink and downlink channels in parallel[1].
- *Bandwidth* The frequency gap between GSM uplink and downlink (in Frequency Division Duplex) is higher than

[1]In February 2013 Ettus released the new Quad Receiver QR210, which integrates four individual front ends but at high cost of several tens of thousands of dollars.

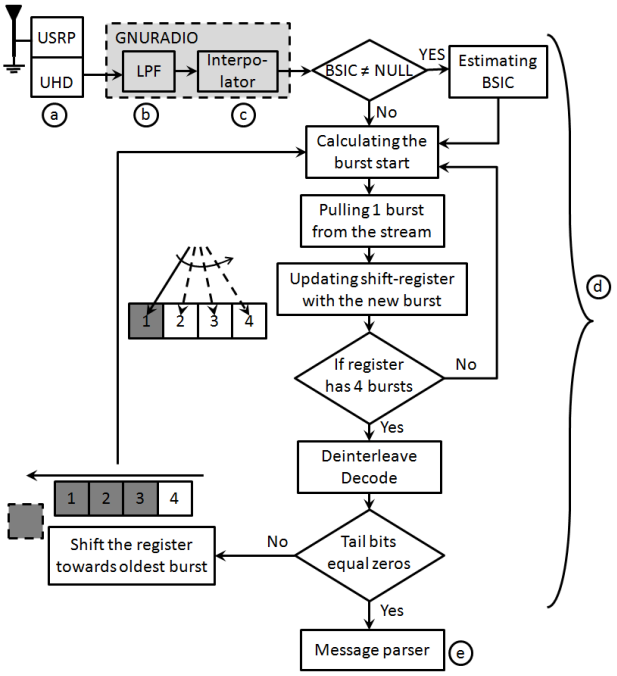

Fig. 4. Algorithm for passive synchronization recovery for an uplink signal.

what the current USRP daughterboard supports, preventing capturing of the complete band by a single front end receiver.

- *Processing Delay* According to our measurements, current latency of retuning in both hardware and software is around 5 ms, which is too high to allow quick switching between uplink and downlink.

It is possible to connect two USRP devices in parallel to scan the uplink and downlink channels but at double system cost. Given the above restrictions, we propose to conduct time synchronization with the MS directly on the uplink, relying on the structure of the Normal Burst. Similar to downlink synchronization, where the long TSC(SB) is used, we are recovering the NB by correlating it to its training sequence TSC(NB).

B. Proposed Passive Receiver

A preliminary step to message recovery is frequency synchronization. This is done by scanning the downlink to create a mapping between the serving cells and the uplink allocated frequencies in that cell. After tuning to an uplink frequency, as illustrated in Figure 4, the captured samples pass from the UHD (step a) to GNU Radio LPF and Interpolator (steps b and c). For time synchronization, the passive receiver applies the following phases on the received samples from the Interpolator: (i) discovery of the used TSC(NB); (ii) NB detection; (iii) message reconstruction and (iv) message parsing. These phases form step d in Figure 4 and are implemented in the 'gsm-receiver' block in Figure 3.

Determining TSC(NB) In Section II-A, we introduced the TSC(NB) as one out of eight training sequences, identified by a sequence number. The TSC(NB) used by a cell is indicated by its number in the cell BSIC. Hence, by listening on the downlink, we can capture the SB and discover the TSC(NB). To avoid the need of retuning between downlink and uplink we propose an alternative, namely cross-correlating (C_{xy}) the received signal stream S with the eight predefined $TSC_i(NB)s, i = 1...8$. The TSC(NB) that results in the highest peak-to-average ratio ($p2a$) of the corresponding correlated stream (and in a successfully decoded message) is chosen for burst recovery. The procedure can be expressed as:

$$\text{BSIC} = \arg\max_i (p2a(C_{xy}(S, TSC_i))) \mid i = 1:8$$

where the $p2a$ ratio is calculated from the peak amplitude of the signal divided by the signal mean value. Finding the TSC(NB) needs to be done only once per cell since the BSIC does not change over time.

Detecting NB After determining the TSC(NB) used in the cell, we can detect its appearance in the uplink signal and thus identify individual NBs. Knowing the TSC(NB) position within the Normal Burst's structure can help us to further determine the burst start. As illustrated in Figure 2, the burst start is an integer shift from the peak of the cross-correlated signal. Once recovered, a complete burst is passed to a demodulation module and the resulting bits are entered into a shift register for message reconstruction. Recall that a single message is carried over four NBs. To align the 1/4 bit at the end of a NB (each NB is 156.25 bits long) we are running a 157-156-156-156 pattern shift as recommended in [6].

Message reconstruction As described in Section II, messages from the data link layer are distributed over four NBs within four consecutive TDMA frames. Typically, a MS knows the correct burst sequence in the GSM multiframe from downlink broadcast information. To overcome the lack of this information in the uplink, we implemented a shift register with the size of four NBs. Demodulated bursts [2] are kept in the register until it is full, upon which event the register's content is sent for deinterleaving and decoding [1]. If an actual message is recovered, it is passed on to the parser. Otherwise, the register drops the oldest written NB and the procedure is repeated with a new NB filling the register. Reconstructing messages of multiple MSs is achieved by updating multiple counters and multiple shift registers in parallel.

Message parsing Finally, we implemented a message parser to enable identification of uplink messages (step e in Figure 4). We need to reconstruct GSM messages to find MS identifiers. The parser follows the 3GPP specification on GSM message formats.

V. Experimental analysis

The implementation and evaluation work was performed using the USRP SDR platform by Ettus. In particular, we used an N210 motherboard, which runs the SDR software on an external notebook (Lenovo T520), and an SBX daughterboard, which contains one receiver front end in the 400-4400 MHz frequency range. The SDR software includes GNU Radio blocks for the basic signal processing and our own modules written in C++.

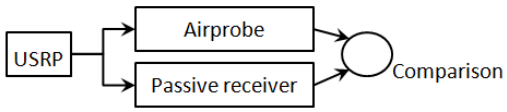

Fig. 5. Downlink evaluation experiment setup.

The processing chain in Figure 4 is tuned as follows: (a) USRP filter bandwidth of 200 KHz; (b) the GNU Radio LPF operates at 135 KHz cut-off frequency and 10 KHz transition bandwidth; (c) the fractional interpolator adapts the sampling rate (300 KHz) to achieve the desired GSM signal rate of four.

Experiments on capturing GSM messages were performed in the city of Bern, Switzerland, on uplink and downlink channels administered by the Swisscom network [7]. In this first version of the code, we focused on single channel capturing. Latest version, however, includes wideband capturing capabilities, which is a requirement for hopping channels and parallel user monitoring. Hopping channels were not relevant for our current study because we capture non-hopping signaling channels.

A. Evaluation of synchronization accuracy

In order to evaluate the ability of the uplink passive receiver to synchronize with the mobile devices we need to compare its performance to a benchmark. We are not aware of any passive receivers for GSM uplink capturing. OpenBTS cannot be used as an uplink benchmark due to spectrum license issues. Hence, we decided to test our method on a downlink channel with Airprobe as benchmark. Airprobe makes use of FB and SB synchronization and was optimized for downlink capturing, reaching message recovery rates of 95%. Our method uses NBs in the downlink for synchronization. We tuned two USRPs, one of them running Airprobe, the other one our passive receiver, on the same downlink Swisscom 939MHz channel and conducted outdoor measurements for two hours in parallel as shown in Figure 5. Table I shows that in best case performance the passive receiver recovers 99% of the messages decoded by Airprobe, with a mean rate of around 98.8%. Combined with the recovery rate of Airprobe, this means our method recovers 94% of the downlink traffic.

The slight imperfections in the observed performance can be caused by imperfect synchronization or poor signal strength. To determine the impact of the latter we compared the received power distribution of both decoded and non-decoded downlink messages. As Figure 6 shows, the two Probability Density Function (PDF) distributions are greatly overlapping, excluding poor signal strength as a cause for missed messages. This leaves imperfect synchronization. First, as shown in Section II-A, TSC(NB) was not designed for synchronization purposes and thus may occasionally lead to wrong calculations of the burst start. Second, wrong shift-register initialization, i.e., missing the first NB of a message, may occur. Although, the performance of the passive receiver could be improved, it already offers a reliable base for GSM signal capturing. There is, however, a distinct advantage with uplink capturing with the passive receiver over Airprobe.

TABLE I
SUCCESS RATE OF PASSIVE RECEIVER

Downlink message type	Passive receiver	Airprobe	Ratio (%)
Paging request type 1	61911	62581	98.8
Paging request type 2	19083	19262	99.0
Paging request type 3	17032	17206	99.0
RR System Info1	1477	1493	98.9
RR System Info3C	2955	2987	98.9
RR System Info4	2957	2986	99.0
RR immediate Assignment	6881	6973	98.6

B. Uplink capturing evaluation

Based on the downlink tests, we expected equally good message recovery in the uplink. However, our in-house tests on uplink channels showed wider RSS variation than on downlink channels. To investigate the impact of this variation, we conducted more detailed evaluations on a Swisscom 894MHz (uplink) channel. We present the findings of an eight hour-long experiment.

Figure 7 plots the PDF of the RSS for decoded and non-decoded messages. The distributions are a combination of messages generated by all MSs transmitting at that channel. Looking at the figure, there seems to be a threshold (around -55dBm, see decoded distribution) below which messages are rarely decoded. Moreover, we can indeed see that the RSS variation in each uplink distribution is much larger compared to the downlink in Figure 6. On the uplink, the transmitters (MSs) have different distances to the passive receiver. In combination with MS mobility and changing multipath propagation, the effect is a variation in RSS. Lower RSS values compared to the downlink measurements are explained by the typically lower transmit power used by mobile devices in contrast to an electricity-powered base station. Furthermore, a passive receiver has less radio visibility to the MSs than a BS and experiences worse propagation conditions. In addition to the RSS effect, the different distances of the MSs may occasionally cause time-overlapping of their messages at the receiver, thus affecting the success of message decoding. Such an overlap is inherited to the system and cannot be compensated.

In an attempt to improve the message capturing rate, we

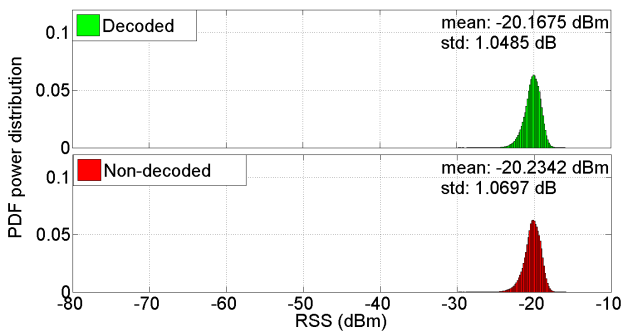

Fig. 6. Distributions of decoded and non-decoded downlink messages.

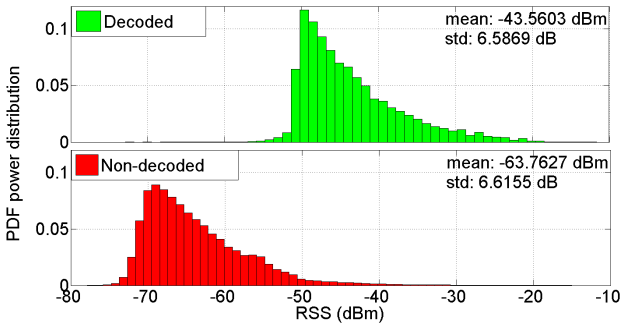

Fig. 7. Distributions of decoded and non-decoded uplink messages.

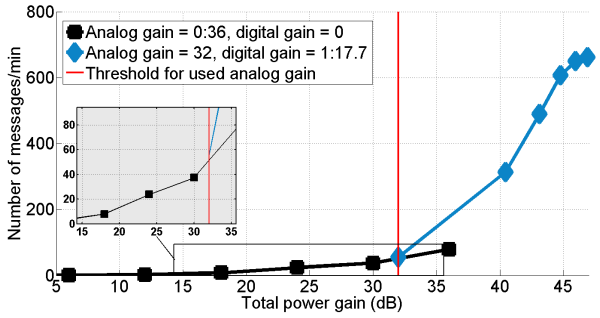

Fig. 9. Power gain relation to the number of decoded messages per cell.

conducted investigations with changing receiver power gain. The total power gain can be seen as the composition of two gain components: analog gain from the hardware (USRP) and digital gain from the software (GNU Radio). First, we gradually increased the analog gain from 0dB to 36dB. Next, with 32dB analog gain, we increased the digital gain up to 17.7dB. Figure 9 shows the number of received messages as the total power gain increases. Although increasing the power gain benefits to message recovery, it should be carefully used, since operating the USRP at maximum power gain for long periods may cause hardware failures. Therefore, we recommend selecting the analog gain close to 90% of the maximum and gradually adjusting the digital gain.

In addition to analyzing power levels, we also evaluated the type of messages that we can recover. We built a GSM parser for most interesting uplink messages. The parser can also detect encrypted messages without decrypting them. We aim to gain insight on the number of unencrypted identified messages, i.e., the messages containing MS identity. Figure 8 shows the identified messages distribution over a working day period. The distribution reflects the known mobile user's activity between day and night.

From the previous set of experiments, our proposed passive receiver showed reliable performance in capturing uplink and downlink messages. The receiver is also able to parse unencrypted messages and extract their MS identity for any further service development. The next research step seeks a wide-band capturing solution that allows us to later retrieve the individual single-band channels.

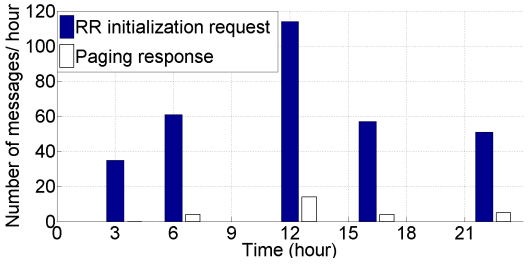

Fig. 8. Unencrypted messages distribution over a working day period.

VI. Conclusions

The paper presented the problem of GSM uplink capturing by third-party devices from a theoretical and a realization perspective. We proposed a new time synchronization implementation that uses the structure of uplink frames to achieve synchronization with the mobile device. The method can synchronize to multiple mobile devices simultaneously by only overhearing uplink traffic. The proposed passive receiver was implemented using SDR Platform. Its performance evaluation showed an average of 98.8% success rate of message recovery. We also investigated the impact of received signal strength on the receiver's performance. The results indicate an RSS threshold, below which successful message decoding is not possible. The limitation from low RSS values was minimized using optimized receiver power gain values. Finally, we implemented an uplink message parser limited to unencrypted messages for later service development. Since the receiver operates independently from the network operator, it is attractive for third parties interested in GSM signal recovery, one of its applications being passive MS localization.

References

[1] Digital Cellular Telecommunications System (phase 2+); Channel Coding (GSM 05.03).
[2] Digital Cellular Telecommunications System (phase 2+); Modulation (3gpp TS 05.04 version 8.4.0 release 1999).
[3] Digital Cellular Telecommunications System (phase 2); Multiplexing and Multiple Access on the Radio Path (GSM 05.02 version 4.9.0).
[4] Ettus Research, A National Instruments Company. [Online]. Available: http://www.ettus.com/.
[5] OpenBTS. [Online]. Available: http://wush.net/trac/rangepublic.
[6] Recommendation GSM 05.10 : "Radio Sub-System Synchronization".
[7] Swisscom. [Online]. Available: http://www.swisscom.ch/.
[8] Ioan Lita et. al. A New Approach of Automobile Localization System using GPS and GSM/GPRS Transmission. *Electronics Technology*, pages 115–119, 2006.
[9] Nisarg Kothari et. al. Robust Indoor Localization on a Commercial Smart Phone. *Procedia Computer Science*, vol. 10, pages 1114–1120, 2012.
[10] Richard Rose et. al. A GSM-Network for Mobile Phone Localization in Disaster Scenarios. *Microwave Conference (GeMIC)*, pages 1–4, 2011.
[11] Stefan Zorn et. al. A Novel Technique for Mobile Phone Localization for Search and Rescue applications. *Indoor Positioning and Indoor Navigation*, pages 1–4, 2010.
[12] Veljo Otsason et. al. Accurate gsm indoor localization. In *The Proc. of UBICOMP 2005*, 2005.
[13] Christian Bettstetter et. al. *GSM - Architecture, Protocols and Services*. Wiley, 2009.