ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (484-494), Month: April-June 2014, Available at: www.researchpublish.com

ACTIVE GSM MONITORING

Prof. Kanaiya Kanzaria[1], Sanjay S.C[2]

[1]Professor, Department of CSE, Reva Institute of Technology and Management,
Bangalore, Karnataka, India
[2]Student (M.Tech, CNE), Department of CSE, Reva Institute of Technology and Management,
Bangalore, Karnataka, India

***Abstract*: 4.1 billion people around the world depend on GSM for at least a part of their day to day communication. Besides communication, more and more additional services - like payment functionality – are being deployed on top of GSM. It has been over 20 years since GSM was designed, and in that time several security problems have been found, both in the protocols and in the originally secret cryptography. However practical exploits of these weaknesses are complicated because of all the signal processing involved and have not been seen much outside of their use by law enforcement agencies. This could change due to recently developed open-source equipment and software that can capture and digitize signals from the GSM frequencies. This might make practical attacks against GSM much simpler. This paper discusses the current state-of-affairs in vulnerabilities of the GSM air-interface, using open-source signal processing. To this end some currently available open-source hardware and software which can be used for signal capturing and some offshoot projects which specifically target GSM are described. Most importantly these include the Universal Software Radio Peripheral (USRP) together with the Gnu Radio implementation for signal capturing and the Air Probe and Open BTS project for handling GSM signals. An in depth view on the functionality of the air-interface of GSM and its security measures are presented and the feasibility of several attacks on the GSM air-interface using the open-source tools is discussed. The monitoring of GSM in real time is restricted only for law enforcement agencies and authorized Government agencies offering them a powerful tool to intercept and track GSM communication**.

I. INTRODUCTION

GSM was developed in the late 1980s and deployed in most Western countries in the early 1990s. Since then GSM has seen an enormous rise both in its coverage and in the number of subscribers. GSM is perhaps the most successful technology of the last twenty years. A survey by the International Telecommunication Union (ITU) showed that by the end of 2008 there were around 4.1 billion people in the world (over 60%) who had a mobile subscription, while over 90% of the world's population lived in a region that at least has access to GSM [1]. According to numbers by the GSM

Association (GSMA) from 2005 the number of cell phones in circulation outnumbered that of personal computers and television sets combined [2]. Internet, its protocols and equipment, have seen a lot of scrutiny over the years. A lot of people more or less understand the TCP/IP stack and tests against different equipment and configurations happen all the time. In the meantime GSM has not received the same level of scrutiny, apart from academic interest in the encryption. Its protocols are public, but mostly unknown and equipment is not tested by its users.

There are several reasons for this lack of interest in GSM security. The specifications are mostly public [3], but consist of around 2000 documents, each between a couple and several hundred pages long. Several books exists that describe the GSM protocols, but they are pricey and often oversimplified or even incomplete. GSM research is also a lot more expensive than Internet research. For the latter all you pretty much need is a network card. But in the case of GSM, equipment capable of demodulating the traffic signals with enough precision was, until recently, very expensive and always proprietary. In short, researching GSM requires much effort and was very costly. However recently Software

Defined Radio (SDR) solutions have appeared in signal processing, that allow a lot of the traditionally hardware operations to be handled by software, opening up this hardware oriented field to more software oriented experts. Especially the emergence of GNU Radio and the Universal Software Radio Peripheral (USRP) has made precise signal processing available to a much larger audience. It is a relatively cheap and fully open-source solution, backed by a large and very diverse community. This SDR solution can handle the modulations and frequency ranges needed for GSM. The security of GSM is paramount for the privacy of 4.1 billion subscribers and its intrinsic security on the air interface - the connection between mobile phone and cell tower - may have just turned to an intrinsic weakness by the arrival of SDR. The fact that GSM has weaknesses is nothing new. The fact that GSM does not use mutual authentication - a mobile phone authenticates itself to the cell tower, but not vice versa - was quickly seen as a problem [3]. Also GSM specifically does not use point to point encryption between callers. It only encrypts the messages while on the air interface. This allows law enforcers to tap conversations in the core of the GSM network. During the development of GSM there was a fierce debate between NATO signals agencies on whether GSM should use a strong encryption algorithm. In particular Germany, sharing a large border with the Soviet empire, was a strong proponent for using strong encryption. France and most other NATO countries were opposed [4]. Both factions proposed a design for the cipher, with the French design finally becoming the A5/1 cipher used in GSM. Once the, originally proprietary, cipher was reverse engineered several weaknesses where found [5, 6, 7]. Because of export restrictions on the A5/1 cipher an even weaker variant, A5/2, was created to be exported to less trusted nations.

Police typically uses so called IMSI-catchers, which can request the IMSI - a number identifying the SIM card inside a mobile phone - of all phones in the vicinity. Using an IMSI-catcher the police can recover the IMSIs of the phones of suspects, who typically use anonymous pre-paid SIMs. With those IMSIs the police can then try to get a warrant for tapping those phones. Besides tapping phone conversations in the GSM back-end, government agencies can also eavesdrop on the air-interface directly. Commercial equipment for eavesdropping on GSM and the aforementioned IMSI catchers are being sold restrictively to government officials [8]. Meanwhile more and more services are being deployed on top of the GSM network, increasing the incentive for criminals to attack GSM. In several countries you can pay for services or products via text messaging. Several Internet banking applications use the mobile phone as an external (out-of-band) channel to verify transactions. Where previously making un-billed calls was the major economic attraction in attacking GSM, increasingly real money can be made.

II. System Design Architecture

The system is designed and developed as Active OFF-THE-AIR GSM Monitoring System. The Surveillance of UM interface (b/w BTS and Suspect's mobile) provides total traffic monitoring including SMS and Voice, of the targeted Mobile. System uses state of art technology to on-line deciphering of A5.1 and A5.2 cipher algorithms. The system comes in a camouflaged case along with Control Laptop and Accessories.

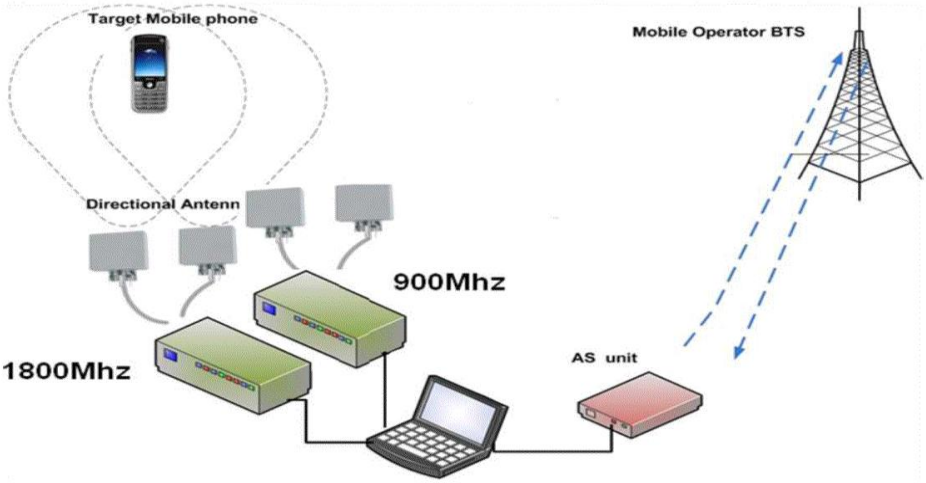

Fig 1. GSM Monitoring System

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (484-494), Month: April-June 2014, Available at: www.researchpublish.com

This system can be used to intercept communications from any GSM service providers in the world irrespective of the type of encryption being used. In case of A5/2, the system is completely passive and does not need any Encryption key info to decipher the communication. In case of A5/1, the system comes with a Kc Retriever to obtain the Kc (Ciphering Key) from the network. As soon as the system identifies that the target communication-taking place is using A5/1 encryption, the Kc retriever in the system becomes active. The Kc retriever automatically acts as a genuine BTS and forces the suspects mobile to register with it, during the periodic location update. The Kc Retriever communicates with the suspects mobile using A5/2 encryption, in the process the system calculates the Kc. The Kc Retriever also asks the mobile to authenticate itself with its IMSI, by doing so the system also obtains the suspect's IMSI. Both the Kc and IMSI information is stored in a database. The Kc Retriever now logoff the suspect's mobile from its BTS. The Suspect is then allowed to communicate with the Actual Service Provider's network. With the Kc in the database the system can now decipher all communications encrypted using A5/1. Alternatively the intercepting can also be performed by forcing the encryption mode to A5/2, thus disabling the encryption mode at the Um interface and communicating in plain text. The system does not require the service providers SIM for operation.

III. GSM Network Architecture

GSM is the most common standard for mobile communication. It is used in more than 200 countries and territories all over the world. The architecture can be illustrated as a hierarchic system of mainly 4 different network components (see Figure 2), which are

- Mobile Stations (MS),
- Base Stations (BS),
- Base Station Controllers (BSC) and
- Mobile Switching Centers (MSC).

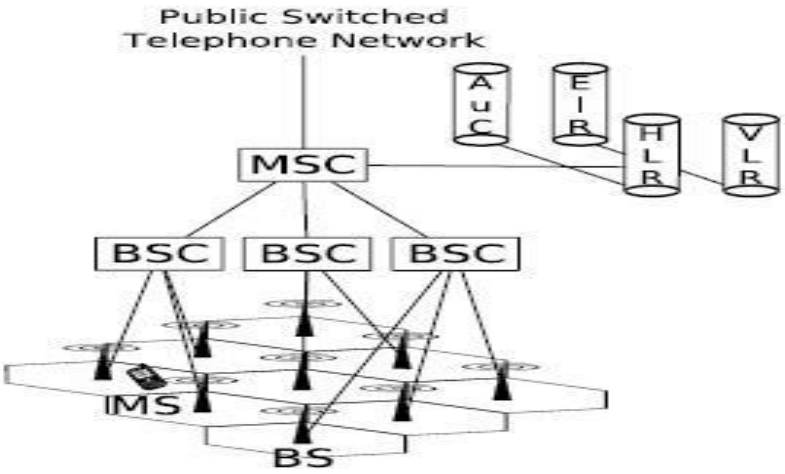

Fig. 2 GSM Network Architecture

In the following we will discuss the functions of these components and the communication between them, to figure out the points of weaknesses in GSM.

A. Mobile Station

A Mobile Station can be seen as a mobile phone with a Subscriber Identification Module (SIM), a removable smart card. Every mobile phone has a unique 15- digit serial number, called International Mobile Equipment Identity (IMEI). In practice, this can be used to prevent stolen phones from accessing a network. The identification of the subscriber is effected with the help of the SIM. It contains the International Mobile Subscriber Identity (IMSI), which is also a 15-digit number, concatenated of

1. The mobile country code (MCC): 3 digits,

2. The mobile network code (MNC): 2 or 3 digits and
3. The mobile subscriber identification number (MSIN): maximum 10 digits.

Furthermore, two algorithms are implemented on the SIM:

• The authentication algorithm A3 and
• The key generation algorithm A8.

For both algorithms, a 128 bit secret key Ki is needed that is also stored on the SIM.

B. Base Station and Base Station Controller

The wireless connection of a Mobile Station and a Mobile Switching Centre is realized by a Base Station1. Therefore, the area is divided into several cellular networks with one Base Station for each cell. The size of the cell depends basically on the geographic features of the area and consequently on the range of the stations. But also the number of possible calls, that have to be handled simultaneously, has to be considered, since it is limited by the number of available channels. Hence, in densely populated areas, the cells often have a diameter of only a few hundred meters, whereas in sparsely populated areas several kilometers are usual.

In subway stations or large buildings Relais Stations are installed to ensure high connectivity. These Relais Stations act like a Repeater in wired networks. They simply amplify and relay incoming signals to the nearest Base Station. However, Base Stations are not only responsible for the connectivity. They are also needed for encryption and decryption of communication data. As the name implies, on the next higher level the Base Station Controller manages the collaboration of the Base Stations and induces power controlling if necessary. If a Mobile Station moves from one cell to another during a call, the Base Station Controller accomplishes a handoff. The connection is transferred to the second Base Station to avoid a termination of the call. The assumption is that both Base Stations are linked with the same Base Station Controller. Otherwise the handoff has to be managed by the Mobile Switching Centre.

C. Mobile Switching Centre

- The Mobile Switching Centre has the role of a mobility management. It is responsible for the authentication, routing, handoffs over different Base Station Controllers, connection to the landline, etc. For this purpose, there are 4 data bases available:

- Home Location Register (HLR): There is only one HLR in one GSM network, which stores personal information's of the subscriber, e.g. the IMSI, the phone number3 or the GSM services.

- Visitor Location Register (VLR): Every MSC has its own VLR. It holds dynamic information's of the subscribers that are under the jurisdiction of the respective MSC. The information's are mostly copies of the personal information's, stored in the HLR.

- Authentication Centre (AuC): The AuC holds the access data of every subscriber, particularly the secret key Ki of the SIM.

- Equipment Identity Register (EIR): As already mentioned, it is possible to prevent mobile phones from accessing the network. To realize this, the IMEI numbers of banned or stolen phones are kept in the EIR.

D. Authentication

If a Mobile Station wants to access a network, a challenge-response protocol is used to authenticate the subscriber (see Figure 3). After sending the security capabilities, the Mobile Station is induced to transmit its IMSI to the VLR. The VLR forwards the IMSI to the HLR and receives a 128 bit random number RAND, a 32 bit signed response SRES and a session key Kc. Only RAND is passed to the mobile station. Together with the secret key Ki, these are the two inputs of

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (484-494), Month: April-June 2014, Available at: www.researchpublish.com

the authentication algorithm A3. As output, the signed response SRES0 is generated, which is returned to the VLR. If SRES and SRES0 match, the authentication request will be accepted, otherwise it will be discarded.

As a security measure, the VLR assigns a Temporary Mobile Subscriber Identity (TMSI) to the Mobile Station to reduce the frequent transmission of the IMSI. This helps to avoid being identified or tracked. In further sessions, this TMSI can be used as identity response.

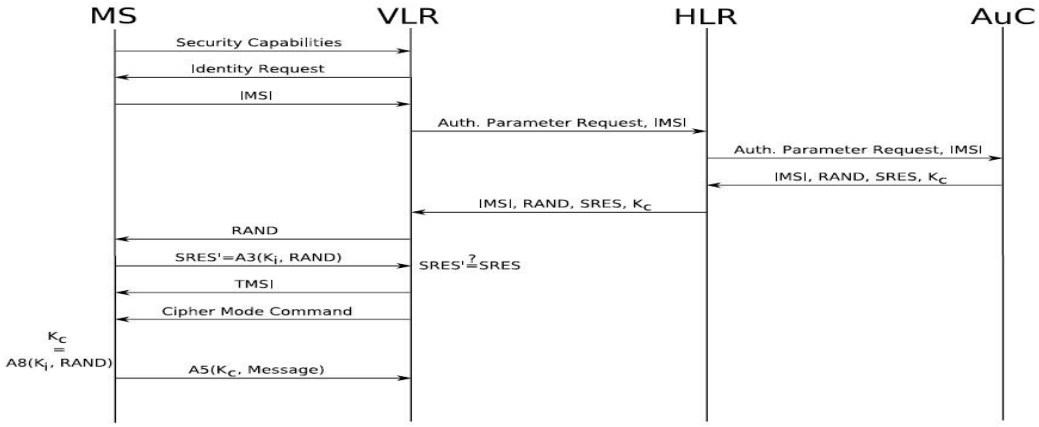

Fig. 3 Authentication of Mobile User in GSM Network

E. GSM Encryption

To provide communication privacy, the stream cipher A5 is implemented on every mobile phone. It is a combination of three (A5/1) or four (A5/2) linear feedback shift registers (LFSRs) which encrypts and decrypts the communication data, if needed. The session key Kc, used in this algorithm, is generated by the algorithm A8 on the SIM with the input RAND. In this case, RAND is the same random number as it is used in the authentication process. As an alternative, A5/0 encloses no encryption at all. In the security capabilities, the Mobile Station specifies, which encryption algorithms are supported. The Base Station chooses one of these and informs the Mobile Station with the cipher mode command.

F. Weaknesses

GSM has a few weaknesses, but regarding to the use of an Active GSM Monitoring System, we will focus on first point.

1. The authentication process considers only a one-sided authentication. The Mobile Station has to prove, that it is permitted to access the network, but there is no verification of the Base Station.
2. Security by obscurity: Since the beginning of GSM, the algorithms A3, A8 and A5 have not been published and always kept secret. But with the help of reverse engineering a number of serious weaknesses have been identified.
3. The encryption is only applied for the wireless transmission. That means, every communication is sent in plain text from the Base Station to the gateways.
4. For technical reasons, it is necessary for a Mobile Station to transmit the current location in short periods to the Base Station. This can be abused to track and record the movement profile of a subscriber.

G. Intercepting GSM Traffic

In GSM, both devices take advantage of the one-sided authentication. As already mentioned, it is not necessary to authenticate a Base Station to a Mobile Station. The Active GSM Monitoring system exploits this weakness and masquerades to a Mobile Station as a Base Station. Additionally, the GA 900 in combination with an own SIM, has the functionality to act like a Mobile Station and to perform a man-in-the-middle attack. Figure 3.3 shows the modified GSM protocol.

Every mobile phone has the requirement to optimize the reception. If there are more than one Base Station of the subscribed network operator accessible, it will always choose the one, with the strongest signal. The Active GSM

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (484-494), Month: April-June 2014, Available at: www.researchpublish.com

Monitoring system masquerades as a Base Station and causes every mobile phone of the simulated network operator within a defined radius to log in. With the help of a special identity request, it is able to force the transmission of the IMSI, instead of a TMSI. A similar procedure can be used to identify the IMEI.

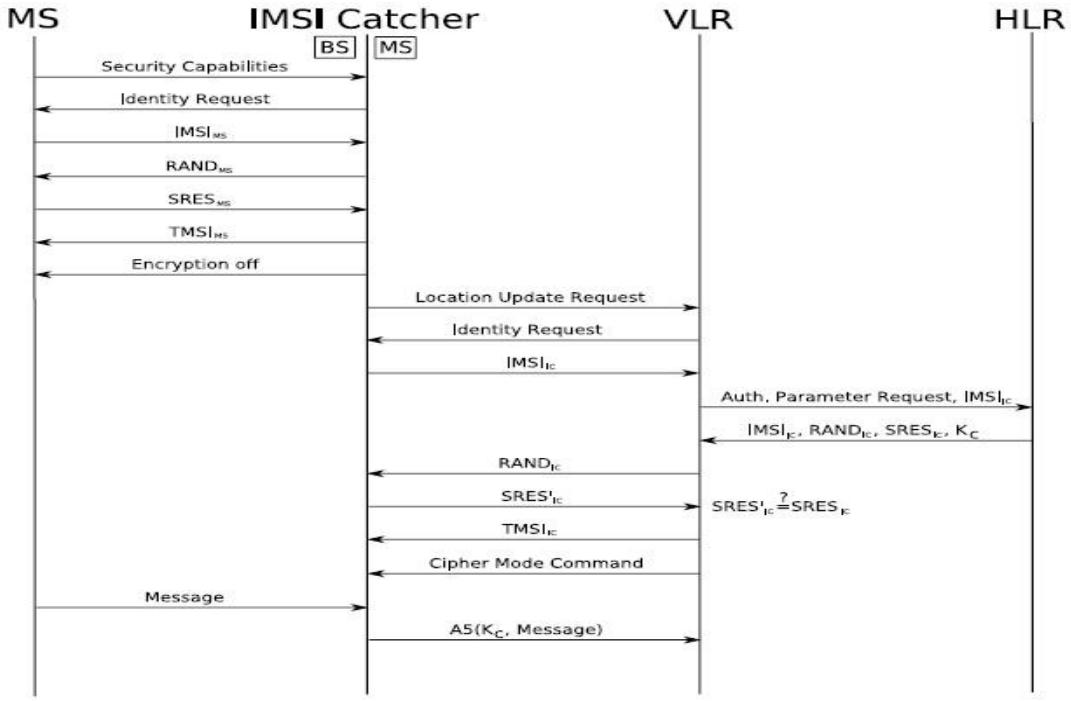

Fig. 4 Intercepting of GSM Network

Figure 4 shows the functionality as communications intercept station of the GA 900. In this case, the Active Monitoring System, which acts as a Base Station, behaves simultaneously like a Mobile Station to the real Base Station. After the authentication process, it uses the fact, that the encryption can simply be disabled from the Base Station. Hence, it can encrypt the plain text traffic from the Mobile Station and pass it to the Base Station.

IV. Hardware and software

This section shows some of the hardware and software that can be used today, to try and sniff GSM traffic. Because the USRP hardware and the GNU Radio software were developed as an SDR implementation, we will first look at the principles of SDR. Then the USRP hardware is discussed and we will look at some of the available open-source software. Finally the exact hardware and software that were used during this research are mentioned.

A. Software Defined Radio (SDR)

Traditionally radios were a hardware matter. They are often very cheap, but also very rigid. A radio created for specific transmit and receive frequencies and modulation schemes will never divert from these, unless it's hardware is modified. Please note that the word radio here is used as a generic transceiver using electro-magnetic waves for transmissions, not specifically as the device known for the reception of programmed broadcasts made by radio stations. The main idea behind Software Defined Radio (SDR), is to create very versatile transceivers by moving a lot of them, traditionally, hardware functions into the software domain. However a radio can never be purely software, because you need a way to capture and create the radio waves. Analog radio waves can be converted to digital samples using a Analog to Digital Converter (ADC) and vice versa using a Digital to Analog Converter (DAC). The ideal SDR scheme involves an antenna connected to a computer via an ADC for receiving and via a DAC for transmitting. All the processing on the signals, like (de)modulation, is then done in software, but the actual transceiving is done in the hardware subsystem. This makes for a much more adaptable system, able to for instance receive GSM signals as well as GPS and also television broadcasts by only changing something in the software. This ideal scheme however is not practically viable, because in practice ADCs and DACs are not fast enough to process a large portion of the spectrum and antennas are designed for specific frequency

bands. This has led to the creation of more extended hardware subsystems for SDRs. Typically such a hardware subsystem consists of a wide band receiver that shifts a frequency band to a standard intermediate frequency, which can be sampled by ADCs and the resulting digital signal can be sent to a computer. Often other common equipment like amplifiers and band-pass filters are also a part of the hardware subsystem. One of the most versatile and widely used SDR systems is GNU Radio, mostly combined with a USRP as the hardware subsystem.

B. USRP

The Universal Software Radio Peripheral (USRP) is designed as a general purpose hardware subsystem for software defined radio. It is an open-hardware device developed by Matt Ettus and which can be ordered through his company Ettus Research [13]. There are currently two types: the USRP1 and the USRP2. Both consist of a motherboard which contains a Field Programmable Gate Array (FPGA), Programmable Gain Amplifier (PGA), ADC(s), DAC(s) and a communication port to connect it to the computer. Daughter boards can be plugged into the USRP motherboard according to the specific frequency bands needed. These daughter boards can be hooked up to appropriate antennas. On the receiving path (RX), a daughterboard captures the required frequency range and sends it through the PGA, possibly amplifying the signal, towards the ADC. The resulting digital signal is passed on to the FPGA, where it is transformed into 16 bit I and Q samples. These are complex samples, with the real part (Q) describing the cosine of the signal, and the imaginary part describing the sine of the signal plus 90 degrees. One sample is thus 32 bit long and can be sent to the host computer through the communication port, for further processing. The FPGA and the host CPU both do some processing on the signal, and though the exact division of labor can be changed, standard the high speed general purpose processing, like down and up conversion, decimation, and interpolation are performed in the FPGA, while waveform-specific processing, such as modulation and demodulation, are performed at the host CPU.

B.1 USRP1

The USRP1 has four daughterboard slots, two for receiving and two for transmitting. It contains four 12 bit ADCs (two for every receive board), that have a sampling rate of 64 Msamples per second. Nyquist's theorem states that you need a sampling rate of at least 2 times the frequency you wish to capture in order to be able to reconstruct the signal [9]. Therefore the USRP1 can capture a bandwidth of 32 MHz at once, for every receive daughterboard. There are also four 14 bit DACs with a sampling rate of 128Msamples per second making the maximum transmit frequency band 64 MHz wide. At the heart of the USRP1 lies it's FPGA, an Altera Cyclone EP1C12. This FPGA can be programmed using the Verilog hardware description language. The compiler for this can be downloaded for free from the Altera website [10]. The communications port is a USB 2.0 chip, with a practical maximum data throughput of 32 Mbyte/s. Since the analog signals are processed into 16 bit I and Q channels, this limits the data throughput to 8 Msamples per second.

B.2 USRP2

The USRP2 contains only two daughterboard slots, one transmitter and one receiver side. It does contain faster and higher resolution ADCs and DACs. Two 14 bit 100 Msamples per second ADCs and two 16-bits 400 Msamples per second DACs. So it can capture a bandwidth of 50 MHz and transmit on a bandwidth of 200 MHz wide. With respect to the USRP1, the USRP2 also contains a much faster FPGA (Xilinx Spartan 3-2000) and an ethernet port instead of the USB connection. The gigabit ethernet port allows for over 3 times higher bandwidth throughput. The USRP2 is much more costly however; double the price of an USRP1.

C. USRP Daughter boards

Different frequencies require different antennas and sometimes different signal processing, like amplifiers or filtering, to receive or transmit correctly. So in order to keep the USRPs as general as possible the actual receiving and transmissions are handled by daughter boards that can be plugged into the USRP motherboard. These daughter boards are specifically meant for certain frequency bands. Currently there are thirteen daughter boards available, of which three are interesting in relation to GSM signals:

• DBSRX, a 800 MHz to 2.4 GHz Receiver.
•RFX900, 800-1000MHz Transceiver, 200+mW output.
• RFX1800, 1.5-2.1 GHz Transceiver, 100+mW output.

The most used GSM frequencies are GSM900 (890.2-959.8 MHz) and GSM1800 (1710.2-1879.8 MHz) in Europe, and GSM850 (824.0-894.0 MHz) and GSM1900 (1850.0-1990.0 MHz) in America and Canada. The DBSRX board covers all these frequencies, but is only a receiver board. In order to actively transmit a RFX board is needed.

D. GNU Radio

GNU Radio [11] is a free software toolkit licensed under the GPL for implementing software-defined radios. It was started by Eric Blossom. It works with several different types of RF hardware, like soundcards, but it is mostly used in combination with an USRP. Basically GNU Radio is a library containing lots of standard signal processing functions. These functions, usually called blocks, are often divided into three categories: source blocks (USRP drivers, but also file readers and tone generators), sink blocks (USRP drivers, graphical sinks like an oscilloscope and soundcard drivers) and processing blocks (like filters, FFT and (de)modulations). These blocks can be attached to each other to make a graph.
All the low level blocks are written in C++, while higher level blocks and GNU Radio graphs are made in Python. These two languages are glued together using SWIG. This means that, for performance reasons, the actual computations are done in C++, while on a higher level a more user friendly language is used to define a software radio. This also abstracts from implementation details for the processing functions. If I want to see a Fast Fourier Transform (FFT) of a certain frequency onscreen, I only need to instantiate a source block (for instance a USRP source, with a frequency) and a graphical FFT sink and link these two together. I do not need to know or understand how the actual FFT is computed, in order to use it. And there are hundreds of implemented blocks inside GNU Radio.

GNU Radio, out-of-the-box, does not offer much in terms of GSM sniffing capabilities, although it can be used to locate the beacon frequencies of GSM masts [12]. However GNU Radio can be used by other software packages, like AirProbe in the next section, to perform the low level functions of GSM sniffing, like reception and demodulation.

E. AirProbe

Airprobe [13] is an open-source project trying to build an air-interface analysis tool for the GSM (and possible later 3G) mobile phone standard. This project came forth out of the GSM-sniffer project [14]. When you currently clone the git repository, you will get nearly ten projects. Some of these serve as libraries for the other projects (e.g. gsmstack), some of these have more or less the same function (e.g. gsm-receiver and T-void) and some of these don't even compile anymore (e.g. T-void). The most interesting part of AirProbe is the gsm-receiver project. It is, at this moment, the best working capture tool for GSM. It comes with two simple shell scripts that call all the necessary functions for saving the signals on a frequency to a file and for interpreting the signals in this file. Calling capture.sh <freq> [duration==10] [decim==112] [gain==52] with a frequency will capture the signals on that frequency to a file. The duration, decimation and gain are optional arguments with default values. A file will be created called capture_<freq>_<decim>.cfile, containing the captured IQ samples. These can then be interpreted by calling: go.sh <file.cfile> [decim==112]

The file name has to be provided, but the decimation is again optional, though you should use the same decimation value that was used during capturing. The go.sh script runs a python file that defines a software radio, which does all the processing needed to get the information bits out of the samples. This results in a series of hex values that represent the information as sent by the GSM network. The go.sh script uses a UNIX pipe method to have these hex-codes interpreted by gsmdecode - one of the other projects in the AirProbe repository. You could also try to convert these hex codes to a .pcap file, which can be read by the wireshark program [15]. Currently the gsm-receiver project will only decode the downlink (GSMnetwork to mobile phone).

F. OpenBTS / OpenBSC

It is useful to know that a Base Transceiver Station (BTS) is a GSM cell tower, and a Base Station Controller (BSC) is a control center for several BTSs. Both of these systems have an open-source implementation: OpenBTS [16] and OpenBSC [17] respectively. This does not mean that both systems work together. In fact they are different approaches to the same problem. OpenBTS, founded by David Burgess, offers a BTS implementation using the USRP and turning it into a BTS. Some of the logic normally present in a BSC is placed inside OpenBTS. OpenBSC, developed by Harald Welte, on the other hand implements most of the BSC functions and currently includes support for two BTS types (nanoBTS and the Siemens BS-11 microBTS). It does not support an OpenBTS driven USRP. Both systems give you the opportunity to run your own GSM network, though this requires a license in most countries. This can be very useful for

testing purposes, but another great use is their implementation of the GSM protocol stack. These open source systems offer an extra way to understand the GSM protocol through their implementation, and these implementations can be used to create GSM analyzers.

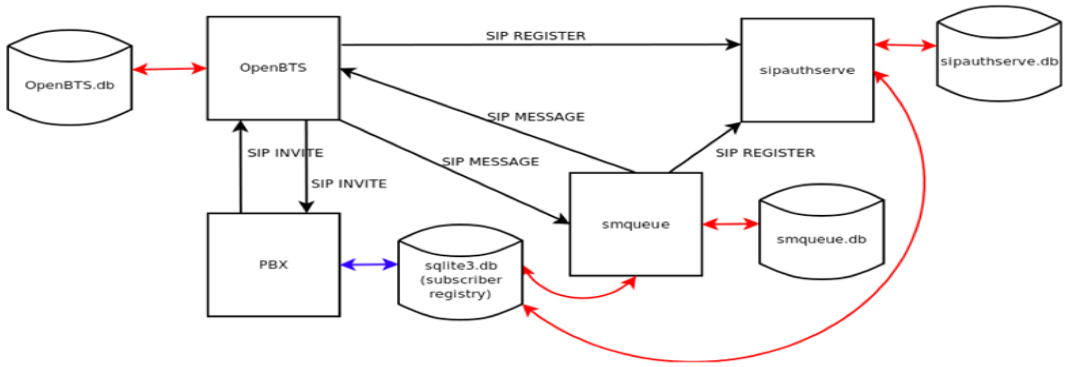

Fig. 5 Flow Diagram of OpenBTS

V. Future Work

A lot of practical work is still needed into the AirProbe project before it becomes a useful GSM protocol analyzer. Firstly it should be improved to actually be able to decode all bursts correctly, but most importantly a solution needs to be found for the hopping problem. AirProbe limits itself to the downlink side of the air interface - cell tower to mobile phone. Capturing the uplink side would be the logical next step, when the down link capturing works. With tools like OpenBTS and OpenBSC, implementing the network side of the GSM system, it is a lot easier to test GSM equipment in a test environment. Similarly it could be very useful to have an implementation of a mobile phone. Recently the OsmocomBB project was introduced, which does exactly that [18]. It is still in early stages, but the use of this project might prove a serious help in GSM research. The flexibility of the USRP and Gnu Radio allow them to be used for research into many different systems. With this relatively cheap equipment it is possible to capture signals from all kinds of wireless protocols, like WiFi and GPS. So other protocols that, like GSM, have seen little practical research because of the difficulties in signal capturing and processing can become open for further inspection. A good example of this would be UMTS. This system is being deployed worldwide and might replace GSM in the long run.

The practical problems with capturing UMTS signals are greater than with the capturing of GSM signals. UMTS's frequency bands are wider for instance, and it employs a more complicated multiplexing technique where different phones communicate at the same time on the same frequency, but their signals are modulated using a different code. So capturing these signals will be even more challenging than capturing the GSM signals. Even so, it would be interesting to judge the feasibility of capturing UMTS signals with the open-source hardware and software. When just looking at the GSM system, it might be useful to research possibilities to increase the GSM security. Naturally GSM's successor, UMTS, is already being deployed and UMTS's security is superior to GSM's - e.g. UMTS has mutual authentication between BTS and MS. However it is unlikely that UMTS will replace GSM completely within the next decade and in the meantime attacks against GSM will only become more feasible. It is of course not easy to adapt the GSM standard without invalidating large quantities of GSM equipment. An approach that could be researched is to replace the fill bits in GSM bursts with a random sequence, instead of the current "2b" encoding. Doing so would remove a lot of the known plain text which is essential for the A5/1 cracking project. A lot of bursts encode the length of their information elements in the second layer header, seemingly removing the need for a standard fill bit pattern.

Naturally the success of this adaptation would depend on the implementation of the GSM stack on most mobile phones. Finally, a subject which mostly falls outside of the scope of this paper, but which will probably become very important, is the security of the mobile phone itself. Modern mobile phones, especially smart phones, are now serious computational platforms. They combine several communication channels; next to GSM and UMTS, phones can also support Blue tooth, WiFi, GPS, a direct link to a PC and an RFID reader. All this increased functionality also makes it easier for an attacker to

try and run malicious code on a victim's mobile phone. It is very likely that for most attackers it will be much more efficient to attack a series of mobile phones through its software.

VI. Conclusion

The GSM system proved hard to understand. Even though most of the specifications are publicly available, the sheer amount of documents and information can be overwhelming. There are many helpful sources to get a broad overview of GSM, but when looking for a more detailed perspective very few sources remain and those that do often contradict each other. In this paper the overview of GSM is by no means complete, but hopefully it will help as a sort of stepping stone for those who want to learn the details of the air-interface of GSM. The publicly available specifications of GSM are in contrast to the extremely closed GSM industry. Only a couple of companies in the world create the core GSM equipment, like the base band processors in mobile phones. All of these implementations are closed-source and often sold exclusively to GSM providers. The immediate effect of this closed nature of the GSM industry is that billions of people walk around with a device, of which hardly anyone knows, or has verified, what it does. Until recently there has only been theoretical research into GSM security. All the strengths and weaknesses of the GSM protocols are more or less known. Now, with affordable and adjustable tools like the USRP and Gnu Radio, more practical research becomes possible so we can actually test the implementations we rely on. Several new attacks have been demonstrated against specific implementations of GSM, like the use of the RRLP protocol to get the location of a mobile phone and disabling mobile phones over the air interface. Implementations of simple theoretical attacks, like building an IMSI catcher are also within reach in the current situation. The more complicated theoretical attacks against the GSM protocol however, are still problematic to implement. Eavesdropping for instance, whether passive or via a man-in-the-middle, is a long way from realization. Though it is feasible, considering the commercial equipment sold for this purpose, there exist a lot of practical problems when trying to implement this using just the open source hardware and software. Most of these problems result from the use of frequency hopping in GSM, which was originally designed solely to improve transmission quality. This is not a plea for releasing a turn-key attack tool, but to give subscribers the ability to verify the workings of GSM, e.g. to check whether, and what kind of, encryption is being used to protect their conversations. The practical problems that at the moment prevent a general attack tool can vary with the specific practical situation. Frequency hopping might not be employed by a specific cell tower, or the cell tower transmits on only a few frequencies that lie close together. In fact a cell tower might not even use encryption. In those cases many attacks become much easier, but we do not know if and how many cell towers have such a configuration. During this research all observed cell towers used both frequency hopping and encryption. It is clear that in the current state-of-affairs the open-source GSM projects have made some attacks more feasible. However the impact of these attacks seems small for now. The open-source GSM projects have not yet directly worsened the confidentiality of conversations over GSM. Despite many recent claims to the contrary no actual conversation has been captured and decrypted, and it will take a lot of effort before the current problems preventing these attacks are solved. It is hard to predict how long it will take the current community behind these open-source projects to solve these practical problems. Though reactions from the community seem eager, the recent rate of development in for instance AirProbe do not show much progress.

REFERENCES

[1] Chris Tryhorn. Nice talking to you ... mobile phone use passes milestone. The Guardian, 2009. Tuesday 3 March http://www.guardian.co.uk/technology/2009/mar/03/mobilephones1.

[2] James Moran. Gsma security group update. In 2nd ETSI Security Workshop: Future Security,2007.

[3] Ross J. Anderson. Security Engineering: A Guide to Building Dependable Distributed Systems, chapter 17. Wiley Computer Publishing, 2001. ISBN: 0471389226.

[4] Technical information gsm system security study. http://cryptome.org/jya/gsm061088.html.

[5] Jovan Golic. Cryptanalysis of Alleged A5 Stream Cipher, page 23955. 1997. http://jya.com/a5-hack.htm.

[6] Marc Briceno, Ian Goldberg, and DavidWagner. A pedagogical implementation of the gsm a5/1 and a5/2 "voice privacy" encryption algorithms, 1999. http://cryptome.org/gsm-a512.html (originally on www.scard.org).

[7] Alex Biryukov, Adi Shamir, and David Wagner. Real Time Cryptanalysis of A5/1 on a PC, page 118. 2000.

[8] Http://www2.rohdeschwarz.com/en/service_and_support/Downloads/news_from_rohde_and_schwarz?issue=152 &type=27&downfileid=1395.

[9] G.W. Gardiner. Handbook of Stochastic Methods, chapter 1.4.4. Springer, second edition.

[10] January 2010. http://www.altera.com/.

[11] September 2009. http://gnuradio.org/trac.

[12] Robert Fitzsimons, January 2009. http://273k.net/gsm/ find-a-gsm-base-station-manually-using-a-usrp/.

[13] January 2010. https://svn.berlin.ccc.de/projects/airprobe/wiki.

[14] September 2009. http://wiki.thc.org/gsm/.

[15] January 2010. http://www.wireshark.org/.

[16] September 2009. http://openbts.sourceforge.net/.

[17] September 2009. http://bs11abis.gnumonks.org/trac/wiki/OpenBSC.

[18] March 2010. http://bb.osmocom.org/.