Master Thesis

Development of a man in the middle attack on the GSM Um-Interface

Adam Kostrzewa

April 15, 2011

Technische Universität Berlin
Fakultät IV
Institut für Softwaretechnik und Theoretische Informatik
Professur Security in Telecommunications

Betreuender Hochschullehrer: Prof. Dr. Jean-Pierre Seifert

Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbstständig erstellt und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Berlin, den April 15, 2011

Adam Kostrzewa

Abstract

Mobile telephony has become an important part of our life, with wide spectrum of everyday applications, also those connected to our financial activities. Therefore it has to be secure and reliable. Unfortunately Global System for Mobile Communications, the most popular standard for cellular communications, has known security weaknesses in ciphering algorithms designed to ensure privacy of data transmitted through radio channels. The ciphers were cryptoanalysed and it was proven that it is possible to break one of them, A5/2, in real time with a ciphertext-only attack.

In the thesis we present elements of the "man-in-the-middle" attack, based on the A5/2 weakness, on the radio connection between mobile and base station. The attack goal is to force a victim's mobile phone to connect to a mobile network operator through the attacker's equipment. In such a situation the attacker can control the transmission. He may eavesdrop, modify, block, or create victim's data.

We designed and implemented the equipment necessary to conduct the attack. Our work contains everything what is needed for the attacker's success from the moment in which victim's mobile phone connects to his base transceiver station to obtaining by him a positive registration result of the stolen credentials. The procedure of victim acquisition was out of the scope of this thesis.

Obtained results showed that attack is possible. Thesis contains the analysis of factors allowing and influencing attack. We present also methods which may decrease attack impact.

Zusammenfassung

Mobile Telefonie und das dazugehörige, weitreichende Spektrum an Anwendungen, ist ein wichtiger Bestandteil unseres Lebens geworden. Bedauerlicherweise existieren bekannte Schwachstellen in den kryptographischen Algorithmen des Global System for Mobile Communications, die ursprünglich dafür gedacht waren, die über die Luft übermittelten Daten zu schützen. Es wurde in der Vergangenheit gezeigt, dass es möglich ist, einen der Algorithmen, A5/2, in Echtzeit mittels einer sogenannten ciphertext-only Attacke zu brechen.

Basierend auf dieser A5/2 Schwachstelle, werden in dieser Arbeit werden die Teile eines man-in-the-middle Angriffes zwischen dem mobilen Endgerät und der Basisstation gezeigt. Ziel des Angriffs ist es, das Telefon eines Opfers dazu zu bringen, über das Equipment eines Angreifers mit dem eigentlichen Operator-Netzwerk zu verbinden. In dieser Situation kann der Angreifer die gesamte Übertragung kontrollieren. Es ist sowohl möglich die Daten des Opfers zu belauschen, zu modifizieren, zu blockieren, als auch ein Opfer zu imitieren. In dieser Arbeit wird das Design und das notwendige Equipment für eine solche Attacke vorgestellt.

Unsere Arbeit umfasst alles, was ein Angreifer benötigt, um nach einer Verbindung mit einer Basiszelle des Angreifers die volle Kontrolle zu übernehmen. Der Vorgang das Opfer dazu zu bringen sich mit der Zelle des Angreifers zu verbinden, wird dabei nicht besprochen, da dies den Rahmen dieser Arbeit sprengen würde. Die gewonnenen Resultate zeigen, dass dieser Angriff möglich ist. Ausserdem präsentieren wir Methoden, die das Ausmaß eines solchen Angriffes abschwächen könnten

Acronyms

SIM Subscribers Identity Module
MS Mobile Station
L1 Physical Layer
L3 Network Layer
BTS Base Transceiver Station
IMSI International Mobile Subscriber Identity
SRES Signed Response
RAND Random Number
Kc Session Key
Ki Individual Subscriber Authentication Key
MCC Mobile Country Code
LAC Location Areal Code
CID Cell Id
RR Radio Resource
MM Mobility Menagement
ME Mobile Equipment
BSC Base Station Controller
SDCCH Standalone Dedicated Control Channel
SMS Short Message Service
BSC Base Station Controller
MSC Mobile Switching Center
PCH Paging Channel
HLR Home Location Register
LAI Location Area Identity
TIMSI Temporary International Mobile Subscriber Identity
SAP Sim Access Profile
VLR Visitor Location Register
AuC Authentication Center
RSEN Remote SIM Enable
ARCFN Absolute Radio-Frequency Channel Number

RACH Random Access Channel
AGCH Access Grant Channel
PC Personal Computer
USB Universal Serial Bus
GPRS General Packet Radio Service
CMUX Converter-Multiplexer
MNC Mobile Network Code
LA Location Area
LAC Location Area Code
GSM Global System for Mobile Communications
USRP Universal Software Radio Peripheral
MNO Mobile Network Operator
BTS Base Transceiver Station
MS Mobile Station
TRX Transceiver
3GPP 3rd Generation Partnership Project
ETSI European Telecommunications Standards Institute
GSMA GSM Association
EIR Equipment Identity Register
OMC Operation and Maintenance Center
IMEI International Mobile Equipment Identity
SPN Service Provider Name
PUK Personal Unblocking Key
FPGA Field Programmable Gate Array
PIN Personal Identification Number
MSISDN Mobile Subscriber Integrated Services Digital Network Number

List of Figures

List of Tables

Contents

1 Introduction

Mobile telephony, with it's increasing popularity, has become an important part of our life. Nowadays we use cells in the wide spectrum of everyday applications. They allow us not only to communicate, what was their original purpose, but also to perform for example our financial activities. With their usage we can pay bills through special premium services, prove our identity to the bank system during online transaction, get an access to emergency services in case of an accident.

The more important they become the more important for us is the security aspect of their usage. We have to be sure that we may rely on them in each situation. Moreover with increasing field of application consequences of malicious actions are becoming more dangerous than ever. An eavesdropped conversation may lead not only to losing confidentiality of it's content but also to immediate and serious financial loses, in case of for example communication with bank system during money transfer operation.

Most popular standard for cellular communication is the Global System for Mobile Communications (GSM) which is used according to the estimations by 80% of the global mobile market and 1.5 billion people in countries all around the globe. It provides a moderate level of security. GSM introduces, to ensure privacy of data transmitted through radio channels, the A5/1 and A5/2 stream ciphers. A5/1 was developed earlier with a stronger algorithm for the usage within Europe and the United States. A5/2 is a weaker version of A5/1 designed for export to other countries. Serious weaknesses have been found in both algorithms.

Marc Briceno reverse engineered their internal design from existing GSM phone in 1999 Soon after that Goldberg, Wagner and Green presented the A5/2 cryptanalysis and first attack. Basing on those achievements authors of the "Instant Ciphertext - Only Cryptoanalysis of GSM Encrypted Communication" document [BBK2007] presented in 2003 the theoretical proposition of a "man-in-the-middle" attack on the GSM Um interface. It's success would allow attacker to take control over whole communication between victim's mobile phone and GSM network and allow wide variety of malicious actions like eavesdropping, identity theft, modification of transmission content.

1.1 Purpose of this thesis

The primary goal of the thesis is to create a practical implementation of the man-in-the-middle attack. During the attack a victim's mobile phone is forced to connect to a Mobile Network Operator (MNO) through the attacker's equipment. In such a situation the attacker can control the transmission. He may eavesdrop, modify, block, or create victim's data. The implementation consist of the three stages of the communication which is happening during the attack: 1) communication between victim's mobile station and attacker's Base Transceiver Station (BTS), 2) communication between attacker's

BTS and attacker's Mobile Station (MS) and 3) communication between the attacker's MS and MNO. It should also contain at least one attack scenario. We selected the "Call theft" scenario. [1].

The second goal is to use in the implementation Open Source solutions and amateur or "low-budget" equipment. Since the moment in which on the market appeared: Universal Software Radio Peripheral (USRP) device and GNU Radio project followed by OpenBTS [2] it became possible even for radio-amateurs to create and setup own GSM networks. That gives the attacker a possibility to conduct malicious actions without access to an expensive custom made equipment and software.

We would like to prove that the man in the middle attack could be done nowadays with relatively small efforts and resources, what makes it more probable.

1.2 Initial assumptions

For purpose of this master thesis, procedures connected with cryptographic aspects of attack, for example breaking A5/2 cipher, were given to the student from the beginning. The author was given access to the libraries designed for breaking the A5/2 cipher which were designed by TU-Berlin worker Dipl.-Ing. Janis Danisevskis. The procedure of the victim acquisition was out of the scope of this thesis therefore the we were allowed to conduct the most important tests with usage of Faraday cage.

1.3 Structure

Below we would like to introduce the structure of the thesis. It starts with Chapter 2 which presents summary of the theoretical information necessary to understand the thesis (such as description of GSM security procedures, design of the A5/2 cipher's algorithm or the conceptual workflow of the man in the middle attack). Chapter 3 presents short summary of the related academic research which became scientific base of the thesis. Chapter 4 describes design assumptions, implementation choices, and technical plans of constructed devices. It includes also a presentation of used in the project software and hardware. In Chapter 5 reader may find the detailed information about the attack's practical implementation, as well as explanation of undertaken actions, with log's examples and charts. Chapter 6 provides the evaluation of the designed equipment. We are trying to assess how dangerous a potential attack can be and what are its main limiting factors. Finally in Chapter 7 we present conclusions with propositions of a further project's development.

1. Full scenario's list is presented in Section 6.4
2. More information about used in the thesis software in Chapter 4

2 Background

This chapter contains a brief review of the most important information about the Global System for Mobile Communications (GSM) infrastructure and security procedures. It will introduce a terminology and technical concepts used throughout the text. We want to give a reader idea about these topics without going into details of related protocols. The chapter starts with a short presentation of the GSM history. Later we provide a description of GSM network's components. This section is followed by a presentation of the security aspects of the standard such as: tools used to provide subscriber's identity (International Mobile Subscriber Identity (IMSI), Temporary International Mobile Subscriber Identity (TIMSI)) as well as authentication and encryption operations. Next we introduce a theoretical background of the man in the middle attack. This attack is the main goal of the thesis. The section about cryptography presents the architecture of A5/2 cipher and the method, used in the attack, to exploit its weaknesses.

2.1 Historical background

GSM is the most popular cellular system in the world. According to the data of the GSM Association (GSMA) in May 2009 there were registered registered 3 milliards of unique subscriber's numbers [1]. That is 80% of the global market. It belongs to the second generation (2G) of cellular technology and it offers a digitized voice rather than an analog as in predecessor systems. Its history starts in 1992 with establishment of the Groupe Special Mobile of the CEPT (Conference Europeenne des Administrations des Postes et des Telecommunications). The acronym GSM was established from first letters of the group name. The goal of the group was to create the one European standard for a cellular communication. Up to that time on the continent there were several incompatible analog networks (e.g. Total Access Communication System (TACS) in the UK, NMT in Scandinavia and the C-Netz in Germany). During the work the group created prototypes of mobile devices as well as conducted research on optimal network access methods. Results of those research became a basis for the formed standard. On the 25th of June 1987 European Council in a directive ordered the member countries to reserve frequencies 890-915MHz and 935-960 MHz for the cellular technology usage. In the year 1989 Groupe Special Mobile became a part of the European Telecommunications Standards Institute (ETSI) [2]. Works on the first standard's version (GSM-900) were finished in 1990 and manufacturers could start to produce the necessary equipment. Later basing on GSM-900 ETSI established the GSM-1800 standard regulating usage of the 1800MHz frequency for the purpose of the cellular communication. First connection in the GSM

1. Statistical data presented by GSMA organization in the [GSM09] report.
2. More information in [ETS].

standard was done in Finland in 1991. In 1992 a first mobile phone was sold to a private customer. From that moment GSM technology rapidly conquered market and remained unbeaten until today.

2.2 GSM infrastructure

In this section we would like to present the list of components which are forming the GSM infrastructure. It is not a complete description of all infrastructure elements but presentation of these parts which are important for the understanding of the thesis, for example there is missing description of the Equipment Identity Register (EIR) and Operation and Maintenance Center (OMC). Graph of the described later infrastructure which is actively used in the thesis is presented in figure 2.1.

2.2.1 Mobile Station

The Mobile Station (MS) is a device that allows us to communicate with a mobile network. It consists of two components: Mobile Equipment (ME) and Subscribers Identity Module (SIM).

ME is a physical mobile phone. It has to operate with a GSM network on at least one from the frequency bands. A quad-band phone will work for us in every place in the world. Each mobile phone has its own International Mobile Equipment Identity (IMEI) number. This number allows identification of the device. It is assigned to a device by its manufacturer.

The SIM is a smart card which purpose is to keep an information about a GSM network's subscriber. Those information later allows his identification during for example the registration procedure. The most important data stored on the SIM card are: IMSI, TIMSI, Individual Subscriber Authentication Key (Ki), Service Provider Name (SPN) , and Location Area Identity (LAI) [3]. On the SIM card an user may keep also for example a list of recently used phone numbers, his phone book entries etc. Those data will be available for him when he will move the card to a different mobile phone. The SIM card is secured by a Personal Identification Number (PIN). If we want to use the card we have to enter the correct PIN which is a four digit combination. Three incorrect attempts to enter the correct PIN block our access to the SIM card content. We may unblock it only with an 8-digit Personal Unblocking Key (PUK).

2.2.2 Base Transceiver Station

The Base Transceiver Station (BTS) is an access point of a MS to the GSM network. Communication interface between the MS and the BTS is known as the Um Interface or the Air Interface. The BTS handles speech encoding, encryption, multiplexing, and modulation/demodulation of the radio signals. It contains of several transceivers, usually from one to sixteen, depending on needs of the particular location. Each Transceiver (TRX) creates a one radio frequency channel which has assigned an Absolute Radio-Frequency Channel Number (ARCFN). A typical BTS covers a 120 degree sector of an

3. More information about them in Section 2.3.

area. That is the reason why transceivers are usually mounted by three. Each device has its own Cell Id (CID) number allowing its quick identification within a particular location. The location has its own Location Area Code (LAC) number. Detailed usage of those numbers is presented later in Section 2.2.6.

2.2.3 Base Station Controller

The Base Station Controller (BSC) is a device which purpose is to control multiple BTS stations. It decides about allocation of radio channels, frequency administration, power and signal measurements, and handovers of a MS from one BTS to another (if both of them belong to the same BSC).

2.2.4 Mobile Switching Center

The Mobile Switching Center (MSC) is a main point of the GSM network. It controls multiple BTS devices and communicates with other MSC stations. It additionally handles call routing, call setup and basic switching functions. MSC also manages the handovers between BSC's and handovers between MSC's.

2.2.5 Home Location Register

The Home Location Register (HLR) serves as a subscribers data base storing following information: phone numbers (Mobile Subscriber Integrated Services Digital Network Number (MSISDN)), IMSIs, current locations of MSs, roaming data, and many others.

2.2.6 Visitor Location Register

The Visitor Location Register (VLR) serves as a base with the subset of the data from HLR limited to subscribers from one Location Area (LA). The Location Area (LA) is a set of base stations that are grouped together to optimize the communication. In such a way the number of queries to HLR is significantly reduced. The VLR is identified by LAC which is a sixteen digit number that identifies a particular LA within the GSM network.

2.2.7 Authentication Center

The Authentication Center (AuC) stores Ki of each IMSI in the GSM network. It also generates variables (Random Number (RAND), Signed Response (SRES) and Session Key (Kc)) used later in cryptographic operations. More details about AuC in Section 2.3.

2.3 GSM security

From the moment of its deployment in 1990s GSM showed that security threats were seriously considered by standard's designers. It introduced secured cryptographic hardware in the mobile station called SIM. System designers postulate was to provide

protection of the air Um interface. In order to do this GSM has to assure privacy of users, that can be accomplished through encryption, and block unauthorized access to the network, for example by a cryptographic authentication of SIM.

All of these procedures are defined in the [3GP97b] standard document. Authentication is done at the beginning of a radio conversation between the mobile station and the network. Whenever a mobile phone tries to access a GSM network it must prove its identity and validity of the SIM card in order to ensure that it is authorized to do that. Later, during transmission, data are encrypted using special cipher algorithm and key (as defined in [3GP97b] section 3.1 and 3.2), to prevent unauthorized access. This process is called encryption.Authentication and encryption rely on the secret key Ki. Copies of Ki are stored in the user's SIM card and in the AuC which can be accessed by a Mobile Network Operator (MNO) from HLR.Ki is encrypted and never transmitted neither by a user neither by a MNO.

A serious shortcoming of the GSM security is the lack of a mutual authentication. The subscriber does not have any means to authenticate the MNO. He can not check if the network to which he wanted to connect to is the one to which he is really connected. That is one of the main GSM vulnerabilities which enable the man in the middle attack. Details are presented in Section 2.4.

The chapter starts with a description of the IMSI and later describes in detail authentication and encryption operations. Information in this chapter is based on the following sources: [BBK07], [EVBH09], [SB11] and [GSM11].

2.3.1 IMSI – TIMSI

The International Mobile Subscriber Identity or IMSI is a unique ID assigned to each user of the GSM network. It is stored as a 64 bit field in the SIM card and it is sent by MS to the network. IMSI has 16 digits and is constructed from: the Mobile Country Code (MCC) which has 3 digits, the Mobile Network Code (MNC) which can have 3 digits but usually 2 digits are used, and the Mobile Station Identification Code (MSIC) which has 10 digits.

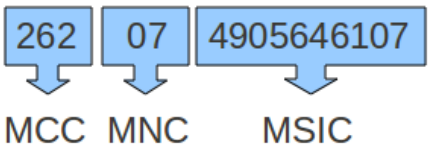

Figure 2.2: IMSI code ctructure

Example of the IMSI code structure is presented in figure 2.2 where: 262 is MCC code of Germany, 07 is MNC of O2 GSM network and 4905646107 is subscriber number (O2 network should have this number stored in subscribers database).

IMSI is the only ID of the subscriber therefore it should be sent through the Um interface as seldom as possible to prevent eventual eavesdropping. In order to achieve this goal after registration in the GSM network, the subscriber will receive a TIMSI.

TIMSI is a pseudo-random number generated from the IMSI number. It is valid in the short range of the GSM network's infrastructure (usually in the range of one LA). In order to track a GSM user via IMSI or TIMSI, an eavesdropper must intercept the GSM network communication where the TIMSI is initially negotiated. In addition, because the TIMSI is periodically renegotiated, the eavesdropper must intercept each additional TIMSI re-negotiation session.

2.3.2 Authentication procedure

The Um interface authentication procedure was described in [3GP98b] Section 4.3.2 and [3GP98a] Section 3.3.1 standard document. It begins when MS send to the network its IMSI code. HLR checks if IMSI from MS is valid and belongs to the MNO. If it does the authentication request is forwarded to the AuC as depicted in figure 2.3.

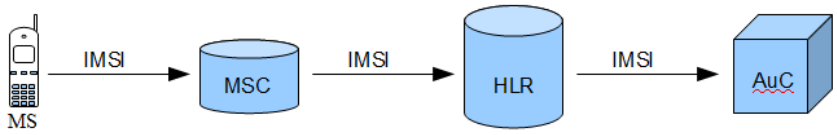

Figure 2.3: IMSI request forwarding to AuC

AuC looks up for Ki associated with given IMSI. It later generates a 128 bit random value, RAND as presented in figure 2.4.

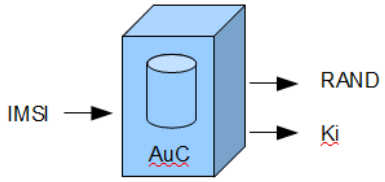

Figure 2.4: Obtaining RAND and Ki from AuC

In AuC RAND and Ki are used as input into the A3 encryption algorithm. The output is the 32-bit signed response number – SRES. RAND and Ki are also inputted into the A8 encryption algorithm. The output is 64-bit Kc.

```
SRES = A3(RAND,Ki)
 Kc = A8(RAND,Ki)
```

Table 2.1: Calculation of Kc and SRES

The Kc is the ciphering key that is used in the A5 encryption algorithm to encrypt and decrypt the data that is transmitted through the Um interface. The procedure is presented in table 2.1.

The RAND, SRES, and Kc are usually known as triplets. The AuC generates many sets of triplets and sends them to the requesting MSC/VLR. This is done to reduce the unnecessary traffic that would appear if the MSC/VLR requested one set of triplets during each authentication of the MS. A set of triplets is unique to the one IMSI. It can not be used with any other IMSI as presented in figure 2.5.

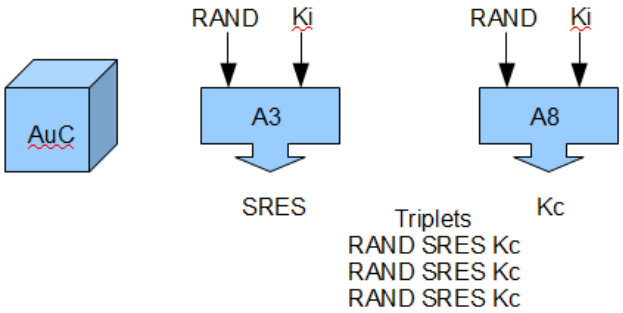

Figure 2.5: Generation of authentication triplets with A3 and A8 algorithms

A set of triplets is sent from AuC back to HLR which forwards it to the requesting MSC/VLR. The MSC stores the Kc and SRES and sends RAND to the MS in the `Mobility Menagement (MM) Authentication Request` message as presented in figure 2.6.

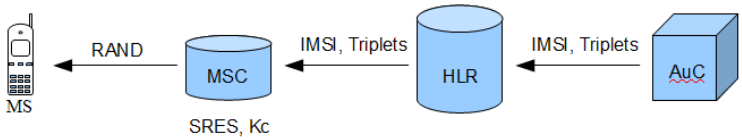

Figure 2.6: Procedure to send a RAND number

The MS calculates a number called SRES by encrypting RAND with an algorithm A3, using Ki as a key and Kc, from RAND and Ki using the A8 algorithm. The MS has the Ki stored on the SIM card. The A3 and A8 algorithms also reside on the SIM card as presented in figure 2.7.

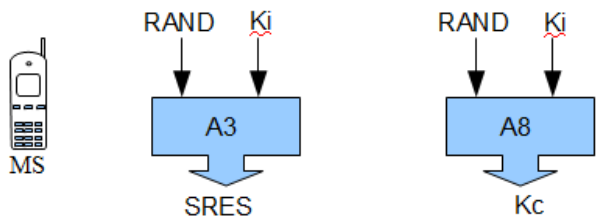

Figure 2.7: Generation of SRES and Ki by mobile station

The MS sends back to the network its SRES in the `Radio Resource (RR) Authentication Response` message. Later the network compares its own calculated SRES with the one obtained from MS and if they match MS is authenticated as presented in figure 2.8.

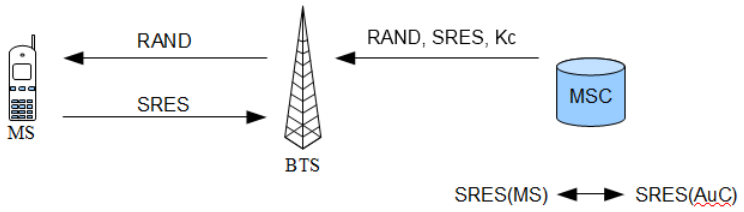

Figure 2.8: Finishing mobile station authentication process

An important fact is that all information which is exchanged between the network and MS during this authentication process can be eavesdropped while encryption is switched on later. The safety of whole process depends entirely on the security and uniqueness of Ki.

2.3.3 Ciphering procedure

The GSM ciphering algorithm is called A5. Ciphering is a radio resource function managed with messages in the Network Layer (L3). It is initiated with an `RR Ciphering Mode Command` message. The command informs which of the A5 variants will be used. The MS starts the ciphering and responds with the `RR Ciphering Mode Complete` message which is sent already encrypted. The encryption is done in the Physical Layer (L1) on the bits of the radio bursts, after forward error correction coding.

There are four variants of A5 in GSM, but only the first three of them are commonly used: *a*) A5/0 - no ciphering at all, *b*) A5/1 - stronger ciphering intended for use in North America and Europe, *c*) A5/2 - weak ciphering, intended for use in other parts of the world, *d*) A5/3 - even stronger ciphering with open design. The network may deny service to any MS that does not support either A5/1 or A5/2. The support for both

A5/1 and A5/2 in the MS was mandatory in GSM Phase 2 ([3GP97a] Section 2) until A5/2 was deprecated by the GSMA in 2006.

Ciphering procedure begins when MSC passes the Kc to the BTS and orders BTS to switch the cipher mode. The Kc should never be passed over the Um interface link as presented in figure 2.9.

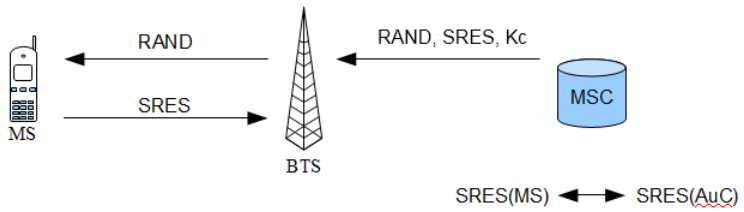

Figure 2.9: Receiving SRES by a GSM network

The BTS inputs the Kc and the data into the A5 encryption algorithm which results in an enciphered data stream. The same procedure is repeated consequently by MS which also inputs the Kc and the data into the A5 encryption algorithm. It is important that the A5 algorithm is a function of the ME not the SIM card as presented in figure 2.10.

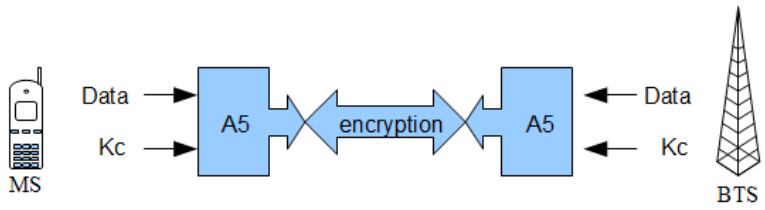

Figure 2.10: Encrypted transmission with usage of the A5 cipher

2.4 Man in the middle attack

The goal of the thesis is to check a possibility of implementation of the man-in-the-middle attack over a GSM's Um interface. In the man-in-the-middle attack a victim's mobile phone is forced to connect to a MNO through the attacker's equipment. The success will give the attacker ability for example to: eavesdrop conversation, hijack call, alter data of Short Message Service (SMS) messages, extend victim's session time, and much more. In this way he gets full control over the connection. When conducted properly, the attack does not present the victim with almost any chance [4] to realize

4. More information about this topic in Chapter 6.

that his connection is being under attack and defend. The man in the middle attack is possible due to two shortcomings of GSM security. The first one is that a subscriber has no means to authenticate the network - there is no mutual authentication. In other words, he cannot check if the network to which he connected is the network to which he really wanted to connect to. Second shortcoming is weakness of the A5/2 cipher which allow the attacker to obtain Kc from encrypted transmission between MS and BTS. Detail description in Section 2.5.

2.4.1 Overview

In this attack we have four types of entities which are interacting with each other. MS belonging to the victim – later in the text simply called as "MS". BTS belonging to the attacker – later in the text called as "fake BTS". MS belonging to attacker – later in the text called as "fake phone" part of equipment which allows the attacker to impersonate a mobile phone in the real network. Real BTS belonging to GSM network – later in text called as "real BTS".

The first attack step is to acquire victim's mobile station. In this phase the attacker tricks the victim's MS to connect to a fake BTS station instead of a real BTS. Later fake BTS authenticates the victim to the fake network. The attacker acquires victim's IMSI. It is used by him when he with a GSM terminal, fake phone, starts registration in the real network and obtains RAND number. He sends to victim's MS the RAND obtained from real network. The victim's MS sends back calculated SRES. Than the attacker forces MS to start encryption and by breaking the A5/2 cipher attacker obtains subsequently Kc. The obtained data are forwarded to the real GSM network. From that moment all traffic between the MS and real BTS goes through the attacker's equipment – the fake BTS and the fake phone.

An important fact is that the private key Ki [5] of the user is never known to the attacker. That allows the attacker to overtake only the current victim's session, which is called a dynamic SIM card cloning. The scheme of the attack is presented in figure 2.11.

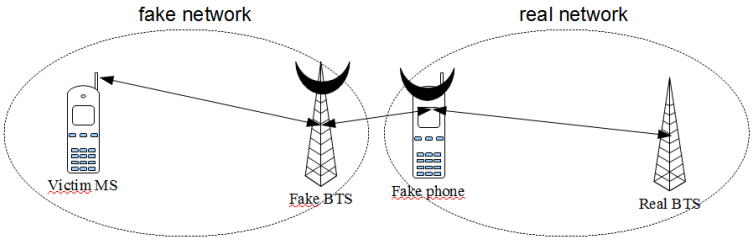

Figure 2.11: General idea of a man-in-the-middle attack

5. Detailed information about it in sections 2.3 and 2.5

2.4.2 Acquiring the victim

In the first phase of the attack the victim's MS is tricked to connect to the attacker fake BTS instead of the real BTS. As stated in the beginning. In this thesis we assume that such connection, between victim's MS and attacker fake BTS, already exists and/or can be established on demand. We present only a theoretical possibilities - practical victim acquisition was out of the scope of the thesis. Information in this section are based on the [Weh09].

2.4.2.1 Setting BTS parameters

In order to fake a real network BTS station we must set up its several identification parameters. Part of them such as LA code LAC, and CID have variable values. They are changing depending on the BTS location. The most important things to consider for faking a real network are two codes which have permanent values: *a*) Mobile Country Code MCC – defining the country in which a particular BTS is placed, *b*) Mobile Network Code MNC – defining the network to which a particular BTS belongs, *c*) Network Shortname – short name of the network in form of a string of characters.

As an example, values of those parameters for most important German operators are presented in Table 2.2.

	T-Moblie	o2	Vodafone	E-plus
Mobile Country Code MCC	262			
Mobile Network Code MNC	01	07	02	03

Table 2.2: Exemplary MCC and MNC codes of german GSM operators.

Another important parameter is the Absolute Radio-Frequency Channel Number (ARCFN) which specifies pair of channels used for transmission (uplink) and reception (downlink) over Um interface. In order to set it up properly we have to gather information about other BTS stations in order not to overlap their signals.

2.4.2.2 Search for the BTS

There are several situations in which MS starts to look for a new BTS station to connect to. We may divide them into scenarios in which the network signal is available and those in which it is not.

In the first case MS does not receive any network signal. It starts then from checking all frequencies used by the BTS stations which were near to the location of the BTS to which MS was successfully connected last time. If none is found it switches to the search mode. In this mode it scans through the standard frequencies in order to find active BTS stations. In this case the attacker's fake BTS must provide the following parameters of the real network: mobile country code MCC, mobile network code MNC, and network short name. This behavior can be triggered through several ways but usually it is done through jamming of the real BTS signal.

In the second case MS is already connected to the network. In this situation there are only two events which may lead to selection of a new BTS. MS may find a BTS station with better signal than one to which it is connected to. In this case fake BTS station is sending signal on the frequency channel used by one of the BTS stations which are near the victim's MS. This scenario is called a forced BTS seclection. A second is a situation in which the frequency channel used by MS in connection with BTS is jammed. In such a situation the mobile station will automatically start to search for a new BTS. This scenario is realized with usage of the Jammer. The jammer is a device designed to send a distortion signal on the given frequency to disrupt the existing connection between MS and BTS.

2.4.2.3 Forced BTS selection

When the MS registers to the BTS, the BTS sends a neighbor list through unencrypted connection. This list contains the frequencies of the BTS stations in the near environment. On those frequencies the MS will receive unencrypted broadcast from the stations and their neighbor lists. Based on this information, the MS selects the BTS with the best signal strength.

Once a fake BTS is switched on and transmits the signal it is still not recognized by the victim's MS. In order to force selection of his BTS attacker can exploit the fact that the MS measures on regular time periods the connection strength to the nearest BTS stations. Knowing frequencies of those connections he may setup the fake BTS to send a signal which will be stronger than any other from BTS stations in the nearest neighborhood. This signal should be send on the frequency of the station with the weakest one. When the received signal will be better than the signal of the existing connection, the MS will change the BTS automatically. This scenario will work only if the MS is in the stand-by mode and no active communication is undergoing. In case there is active communication, we are forcing the handover of this connection within the network. That cannot be done without access to MNO's BSC.

Figure 2.12 shows an illustration of exemplary scenario. In this scenario MS is connected at the beginning to the BTS one. MS knows about the nearest BTS stations, particularly: BTS two, BTS three, and BTS four. The attacker is checking the frequency channel of each BTS station. Later when he will switch on his fake BTS it will start to send a signal on the same channel as the BTS four, which had during the attacker's measurement the worst signal quality. The MS will notice that the quality of the signal from BTS four has improved and switch into it. In such a way victim's mobile station has been lured to connect to the attacker's owned GSM network.

2.4.2.4 Signal jamming

Using this technique, the attacker will first send a distortion signal on the frequency of the existing connection between the victim's MS and BTS. If these distortions are strong enough, the connection will be broken. Later, the MS will automatically start BTS search procedure in order to find a substitute for the jammed one. In such a way it may select the attacker's BTS but it may also happen that the signal from another

BTS will have a better quality and the MS will select it. In order to be sure that the selection's result will be a fake BTS, the attacker may try to jam for a short while the signals from the other closest BTS, based on the neighbor list. The biggest advantage of jamming all nearest BTS stations is that we may force a situation in which the MS will go into a BTS signal search mode and start scanning all frequency channels, not only the neighbor list ones. As a result, the attacker may lure the MS to connect to a fake BTS on any supported by the phone ARCFN channel. That will give him much better connection quality.

2.4.3 Authentication and encryption

The next phase after luring the victim to connect to the fake BTS is conducting the authentication and encryption process. The schematics of information exchange in case of proper registration are presented in figure 2.13. In case of the man in the middle attack, the information flow (for example connections between BTS, VLR, HLR and AuC) on the GSM network side is not changed as proposed in the [BBK07]. Between MS and BTS appear attacker's equipment – namely the fake phone and the fake BTS. The detailed scenario is presented in figure 2.14. It begins when the victim's mobile station is lured to connect to an attacker's fake BTS instead of a real one. In the next step the attacker gets victim's IMSI during the Location Update Procedure and forwards it to the fake phone.The attacker's fake phone is used to impersonate the victim's phone in the real network. In the following step, the attacker starts registration to the real BTS using a fake phone. This invokes a network authentication procedure. The authentication request from a real network, containing the RAND number is forwarded immediately after receiving from the fake phone, through the fake BTS, to the victim's mobile phone. The victim computes the SRES, and returns it back to the fake phone. After that the fake BTS orders the victim's MS to start encryption using the A5/2 cipher. That is normal and legitimate behavior while the attacker impersonates a real network. During the encryption procedure the attacker uses known ciphertext attack to find session Kc, what usually takes less than a second. Obtained Kc, similarly to SRES, if forwarded to the fake phone. The fake phone right now has both values and can finish the authentication and encryption procedure with real MNO. The attacker sends back SRES and later when the network asks (the "authenticated" attacker) to start encryption using the A5/1 the response can be sent encrypted since he already knows the Kc.

2.5 Cryptography

In order to ensure privacy of data transmitted through radio channels GSM introduces the A5/1 and A5/2 stream ciphers. A5/1 was developed earlier with a stronger algorithm for usage within Europe and the United States. A5/2 is a weaker version of A5/1 designed for export to other countries. Serious weaknesses have been found in both algorithms. Marc Briceno reverse engineered their internal design from an existing GSM phone in 1999 [BGW99]. Soon after that Goldberg, Wagner and Green presented the A5/2 cryptanalysis and first attack [GWG99].

The man-in-the-middle attack presented in this thesis is based on weaknesses of the A5/2 cipher. Because of that in this chapter we are presenting short description of the A5/2 cipher based on information from the Barkan, Biham and Keller [BBK2007] document. In order to break the A5/2 cipher I used an implementation of the known-plaintext attack. This implementation was written by Dipl.-Ing Janis Danisevskis based on [BBK07]. After breaking the cipher, the attacker has access to the Kc used to encrypt the connection.

2.5.1 Architecture of the A5/2 cipher

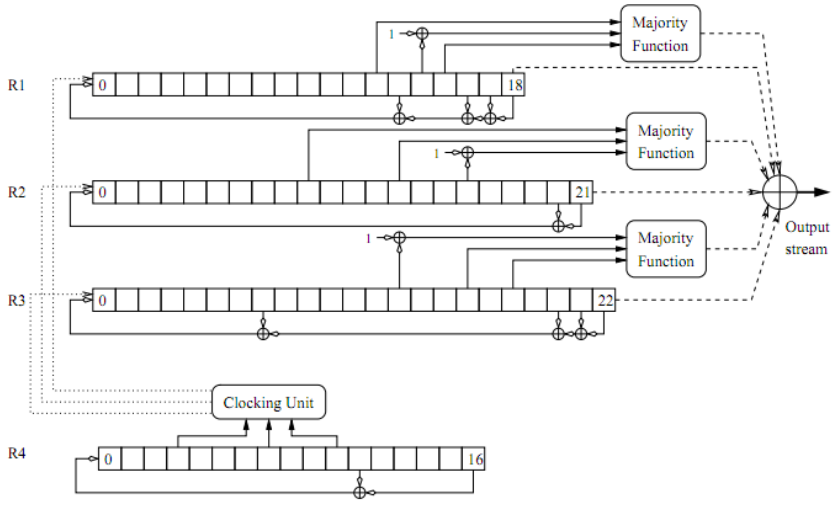

Figure 2.15: Internal structure of A5/2 taken from [BBK07]

The ciphering procedure starts from setting up of the 64-bit key session key Kc and 22-bit publicly known index value denoted by f which changes for each frame. As presented in figure 2.15 the cipher's internal state consists of four registers R1 (19-bit), R2 (22-bit), R3(23-bit) and R4(17-bit) with linear feedback. A5/2 is initialized during so called "key setup" with Kc and f as presented in figure 2.3. Cipher works in cycles. The goal of each one of the cycles is to produce the one output bit. The main idea is that in every cycle two, or three from R1, R2, R3 registers are clocked basing on the values of the three bits from the R4 register. Later R4 itself is clocked. Output bit is a quadratic function of R1, R2 and R3. First output 99 bits of the output are discarded and the following 228 bits create the output keystream. The first half (first 114-bits) of this keystream is used for encryption of connection between a GSM network and the phone. The second half (following 114-bits) to encrypt communication in the opposite direction - from the phone to the network.

The whole process can be summarized in three steps : the initialization of the A5/2 cipher with Kc and f, discarding the output of the first 99 cycles and gathering the output of the following 228 cycles to create the keystream.

1. Set $R1 = R2 = R3 = R4 = 0$.
2. For $i = 0$ to 63
   - Clock all four registers.
   - $R1[0] \leftarrow R1[0] \oplus K_c[i]$; $R2[0] \leftarrow R2[0] \oplus K_c[i]$; $R3[0] \leftarrow R3[0] \oplus K_c[i]$; $R4[0] \leftarrow R4[0] \oplus K_c[i]$.
3. For $i = 0$ to 21
   - Clock all four registers.
   - $R1[0] \leftarrow R1[0] \oplus f[i]$; $R2[0] \leftarrow R2[0] \oplus f[i]$; $R3[0] \leftarrow R3[0] \oplus f[i]$; $R4[0] \leftarrow R4[0] \oplus f[i]$.
4. Set the bits $R1[15] \leftarrow 1$, $R2[16] \leftarrow 1$, $R3[18] \leftarrow 1$, $R4[10] \leftarrow 1$.

Table 2.3: Key setup of A5/2 from [BBK07]

2.5.2 Cryptographic aspects of attack

Below one may find a short introduction based on the [BBK07] document which presents the basic principle of this attack. It is the second part of the introduction presented previously in Section 2.5.1.

In order to successfully conduct attack we need four data frames. We use them later in mechanism of internal state (R1, R2, R3, and R4) recovery which leads to reversing key setup and in the result to obtain the Kc. We start from trying all possible 2^{16} combinations of the R4 state. Each of them gives us system of linear equations. Those equations describe output bits of the corresponding four frames. We have to solve each of the equations independently in order to obtain a proposition of the appropriate internal state (R1, R2, R3). These internal states together with R4 form a candidate of the full internal state. After that operation we may have several candidates of the full internal state from which we have to select one. In order to do it we firstly get rid of those which are inconsistent in the Gaussian elimination. This elimination operation should be very easy and fast to conduct. In the result we will have either already the only one candidate state which will be already our result, either two or more consistent internal states. In case of the second situation we have to eliminate unnecessary ones by trial encryption. Full description of the algorithm is presented in [BBK07] in Chapter 3.2.

This attack can be further optimized, in order to decrease the time of obtaining the session key. The time necessary to achieve our goal can be reduced from minutes in case of regular known-plaintext attack to few milliseconds. That allows us to use it in the man-in-the-middle attack, where the attacker has overall 7 seconds to send back to the GSM network the response coded with the session key. Optimized version in principle works in the following way that before we start the elimination of the unnecessary full internal states we are computing the dependencies which will appear later on. Rather than by check the consistency of the Gaussian elimination we filter for

the R4 by applying consistency check on the know precomputed values. This approach needs precomputed data tables stored in the memory. Detailed description is presented in [BBK07] in Chapter 3.3.

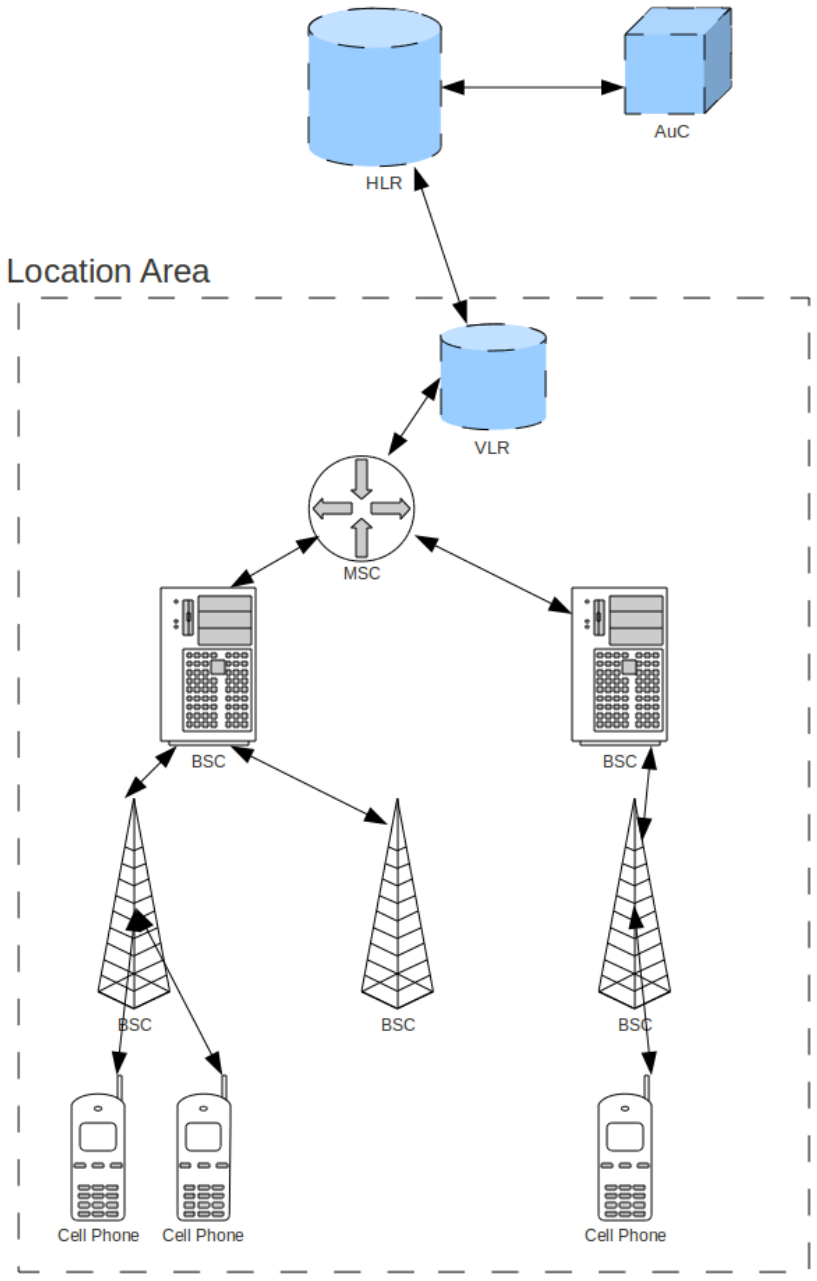

Figure 2.1: GSM infrastructure

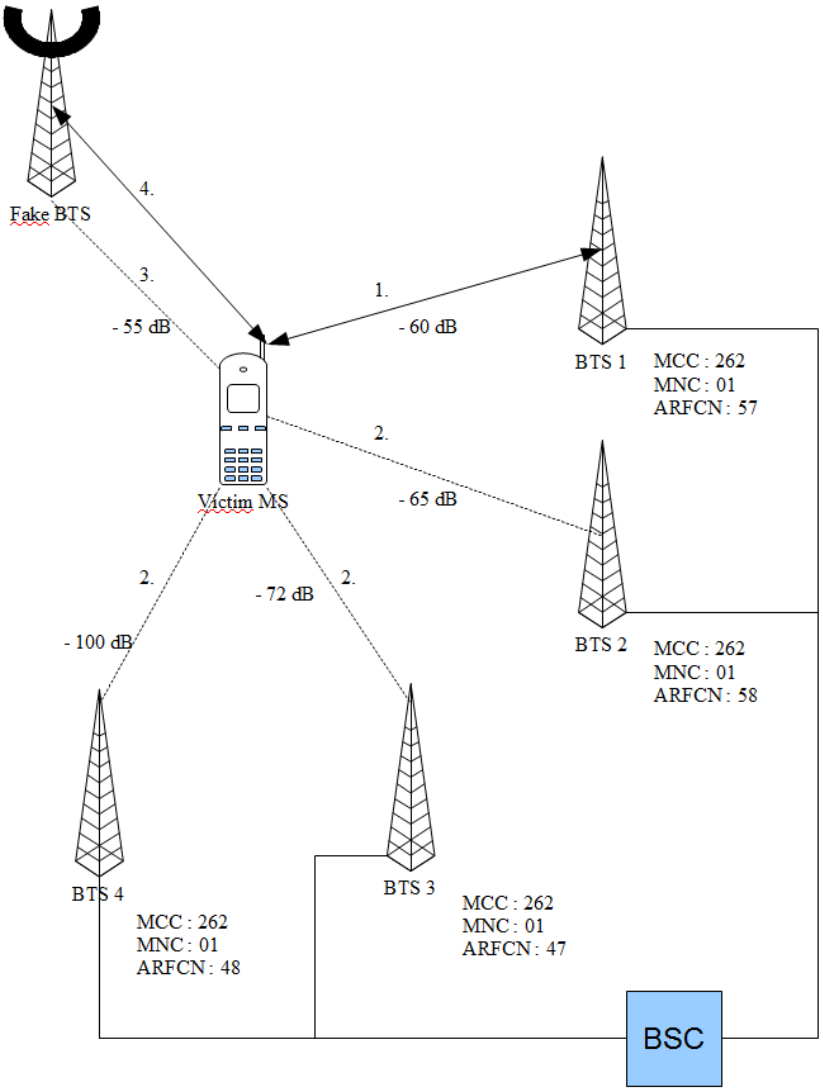

Figure 2.12: Forced BTS selection scenario. Figure based on [Weh09]

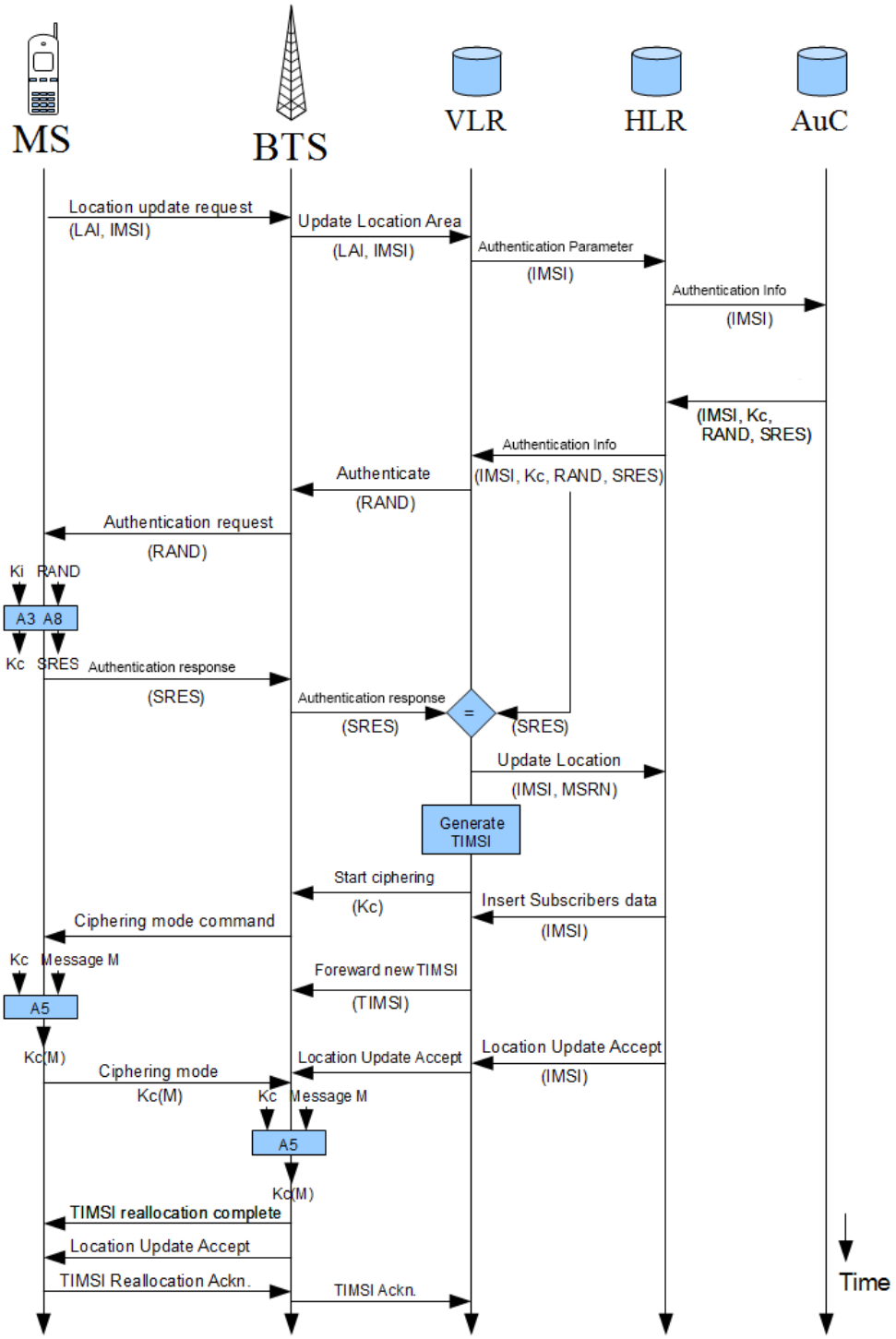

Figure 2.13: Graph of standard procedure of mobile station registration in GSM network based on [EVBH09]

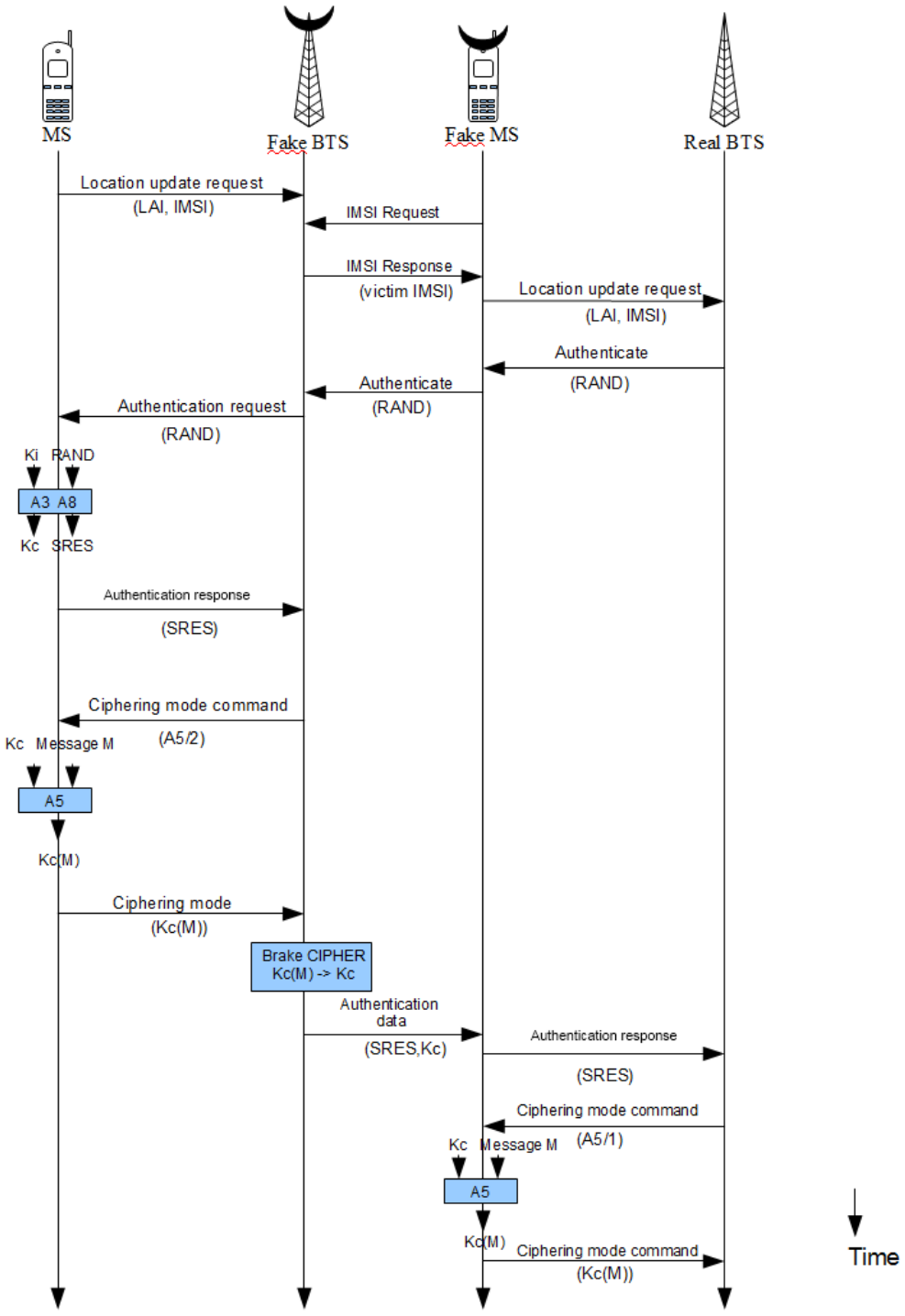

Figure 2.14: Graph of mobile station registration in GSM network in case of man in the middle attack

3 Related Work

In this chapter we would like to provide an information about the previous research done on those aspects of the GSM security which can be used in the man-in-the-middle attack. We will start with an overview of the scientific work about cryptographic aspects of GSM. First section will concentrate on works connected with the A5/2 cipher – its design and cryptoanalysis. In the thesis we check practical possibility of extending the A5/2 attack on the A5/1 cipher as proposed in [BBK07] basing mainly on open source solutions [1]. Similar research does not exist to our knowledge.

In the second part of the chapter reader may find information about research which led to the creation of IMSI catcher.

3.1 Cryptography

The internal design of both the A5/1 and the A5/2 was reverse engineered, for the first time, from an actual GSM phone and presented in [BGW99] in 1999. It was provided in form of the algorithm in C source code verified against known test-vectors.

Very short after that, in the same year 1999, in [GWG99] authors Goldberg, Wagner and Green presented the first cryptoanalysis of the A5/2 cipher which proved that the algorithm is very week. They observed, between the others, that there exist an internal state of the cipher which is forced to have value one during initialization, regardless of the appropriate input data value. Basing upon that they constructed the attack which requires only two data frames with a known plaintext to successfully recover the the Session Key (Kc). Only problem was fact that those data frames have to be transmitted exactly 6 seconds apart. Low encoding complexity allows to conduct the attack in the real time even on a low end machine. All necessary operations can be done in few minutes.

In the next year (2000) Slobodan Petrovic and Amparo Fuster-Sabater proposed improved version of that attack in [PFS00]. They presented the algorithm which can determine a linear relation between output bits. Later, basing upon this relation, it reconstructs the remaining part of the message. The attack is done by construction of the set of quadratic equations with cipher internal states as a variables. The improvement is that attacker needs four data frames with known plain text, but he does not need to wait 6 seconds, which was proven to be an unrealistic assumption. As in case of the previous attack also here the Individual Subscriber Authentication Key (Ki) is still unknown but successfully recovered session key allows to decrypt whole remaining communication.

1. Detailed information about A5/2 ciper itself and used in the thesis know-plaintext attack one may find in section 2.5

Finally in the year 2003 Barkan, Bian and Keller in [BBK07] showed a practical ciphertext only cryptoanalysis that allows to recover a session key basing on the data from several miliseconds of the encrypted GSM communication on regular PC in less than a second. It is based on the observation, similar as in [PFS00], that having a cipher-text only one can find a linear relation between output bits. In case of the known-plaintext attack we know the keystream bits, while in case of the ciphertext-only attack we know the value of linear combinations between them. Authors proved that in practice instead of four data frames (as in case of the known-plaintext attack) we need eight ciphertext data frames to success. In the publication authors show various possible active attack scenarios on GSM protocols. They also proposed a way of extending the attack on the A5/1 cipher. The proposition was in form of theoretical concept without practical implementation. This thesis provide the practical implementation of that idea.

3.2 IMSI catcher

IMSI catcher is a device which creates owned by an attacker GSM network, and taking advantage of lack of mutual authentication in GSM [2], forces victim MS to connect to it. First IMSI catcher appeared on the market in 1996 [3] only five years after the official start of the first GSM network. It was created by German company Rohde & Schwarz in Munich as the "GA 090" device [4]. It is presented in figure 3.1.

Figure 3.1: Rohde and Schwarz IMSI catcher GA 090 taken from [Bil11]

At the beginning officially sold IMSI catchers offered functionality limited to localization of MS within given area. They were able to do it by retrieving of IMEI and IMSI of

2. Detailed description in Section 2.3
3. More information `http://www.iwi.uni-hannover.de/lv/ucc_ws04_05/riemer/literatur/imsi-catcher.htm` [Online, last check 37.03.2011]
4. `http://www.rohde-schwarz.com/Homepage` [Online last check 27.03.2011]

MS stations from particular location. Along with advances in the cryptographic attacks against used by the standard ciphers more sophisticated devices appeared on the market. IMSI catcher was a first step in conducting man in the middle attack. Devices such as "SCL-5020SE" from the indian Shoghi Communications company [5] designed intentionally for a "Police and Public Service" [6] which offers possibility to: eavesdrop conversation encoded with A5/2 as well as A5/1, intercept SMS messages, record transmitted data and much more. Access to those type of the devices is however officially limited to the governmental institutions. Even prices of the equipment are not published officially, but it is advertised as an expensive hardware for professional usage.

This situation start to change with advances in the Open Source software designed for a GSM technology. Last year on the Defcon 18 [7] conference Chris Paget during presentation "Practical Cellphone Spying" [8] presented fully Open Source implementation of the IMSI catcher which price should establish round 1500 Euros [9]. This achievements become a basis for the attempt to create the full man in the middle implementation presented in this thesis.

5. Company website `http://www.shoghi.co.in/communication-security.php`, [Online last check 16.03.2011].

6. More information at `http://www.shoghi.co.in/communication-intelligence.php` [Online last check 16.03.2011].

7. More information about conference at `https://www.defcon.org/html/links/dc-archives/dc-18-archive.html`, [Online, last check 21.03.2011].

8. Presentation available in video stream at `https://www.defcon.org/html/links/dc-archives/dc-18-archive.html#Paget`, [Online, last check 21.03.2011].

9. Information from `http://www.heise.de/security/meldung/IMSI-Catcher-fuer-1500-Euro\-im-Eigenbau-1048919.html`, [Online last check 21.03.2011].

4 Design

In a practical implementation of the man-in-the-middle attack we use two devices. The first one is a fake BTS which allows an attacker to create a fake GSM network. It is made from the Universal Software Radio Peripheral (USRP) [1] device connected to a computer with the OpenBTS software [2] . We modified both the USRPand the OpenBTS to adjust them to the needs of the project. The second component of the implementation is a fake phone. The device is build from a Telit GT864-PY GSM Module connected to a computer and controlled by a script written in Python. It is also using, in order to increase the reliability and development speed, the Sim Access Profile (SAP) server from the Samsung Corby mobile phone. The fake BTS and the fake phone communicate with each other through an Ethernet connection. The communication between a fake BTS and a victim's mobile station as well as between a fake phone and a GSM network is done with the usage of the Um interface. In figure 4.1 we presented a scheme of the equipment's setup. This chapter contains the detailed description of all components used in the project. Detailed description of the attack itself is presented in Chapter 5. First we present the development goals followed by the implementation details and their consequences. The schematics presenting the equipment setup is depicted in figure 4.1.

4.1 Fake Phone

The fake phone, as described in Chapter 2.4, is an equipment which should give an attacker the possibility to use stolen victim's credentials to register and communicate through the GSM network. It has to include a full mobile station module which should enable access to the standard services available in a GSM network, for example voice calls, SMS messages, etc. Those services are present right now in most of the mobile phones on the market. Moreover the fake phone should give an ability to communicate and synchronize actions with a fake BTS. This communication should allow an exchange of the data necessary to conduct a successful authentication and encryption with MNO. During next phases of the attack the fake phone will receive a transmission data from the victim's mobile station. This traffic has to be forwarded to the network, otherwise attack can be discovered by a victim. Along with this operation an attacker may also modify victim's data what increase the spectrum of possible for him malicious actions.

In figure 4.2 we present the diagram of the fake phone work scenario. At the beginning the attacker activates the device. From that moment the fake phone waits for the attack command containing a victim's IMSI code. This command should be send from the fake BTS through the ethernet connection. When it arrives the device activates the GSM

1. Detailed description in Section 4.2.1
2. Detailed description in Section 4.2.2

mobile station module. Inside of this module there is a SIM card. Later the attacker will modify commands and responses to them send by the module to the card. [3]

In the next step MS module issue the command to obtain an IMSI code stored on the SIM card. The response to this command is modified by substitution of the IMSI code with the one belonging to the victim, which has been received at the beginning from the fake BTS. After that the attacker can start process of the module's registration to the GSM network. MS should receive a RAND number which will immediately forwarded to the fake BTS. Then in the response the fake BTS will send back the SRES number and the Kc key. Both of the numbers were obtained from the victim and would be used in the normal connection. Those values, after modification of the appropriate response, are send back to the MS module which completes the registration to the GSM network with stolen credentials.

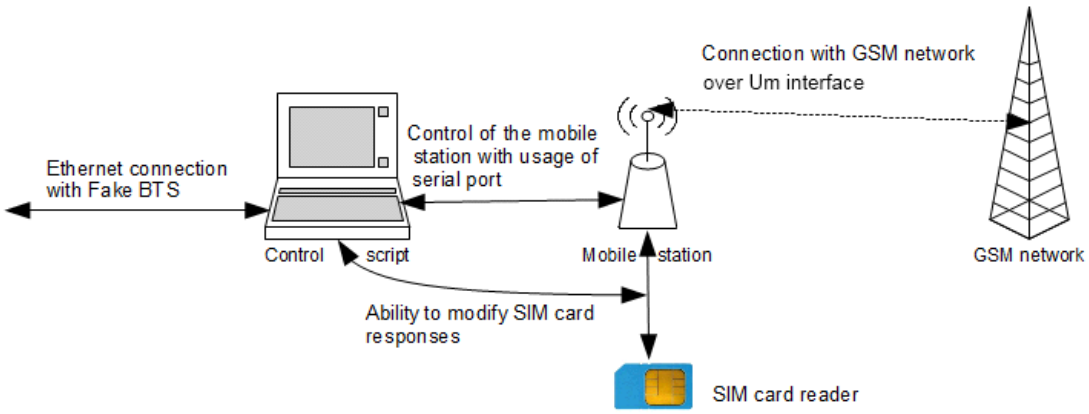

Figure 4.2: Fake phone - the diagram of work

4.1.1 Hardware

In the practical fake phone's implementation we use three main hardware components. A regular low end Personal Computer (PC) with Universal Serial Bus (USB) and Ethernet interfaces. Its goal is to parse commands, modify them, organize the attack in time, and control all devices. The mobile station module which communicates with a real GSM network. Additionally the MS should be remotely controlled by a PC. Finally a SAP server or a SIM card reader. These components are necessary to avoid the creation of a SIM card simulator. During boot procedure the mobile station executes multiple commands, not only those connected with authentication, encryption and reading IMSI. Writing a SIM card simulator or creating a dedicated device is a long and error prone process. It is much easier to redirect whole traffic to some existing SIM card modifying

3. Otherwise we would have to write SIM card emulator which was not the goal of the thesis. More information later in this section.

only crucial for the attack commands. Using of the SAP protocol allowed us additionally to bypass SIM card's time access restrictions which are switched off during the server – client communication. The detailed description is provided in Chapter 6.1. In order to avoid an implementation of a SAP server, we used the SAP server from the Samsung's Corby mobile phone.

4.1.1.1 Mobile Station

Figure 4.3: Telit GT864-PY mobile station's front and back take from [Com08]

In our project we decided to use the Telit GT864-PY GSM mobile station module [4] produced by the Telit Communication [5] company. It fulfills all our requirements. It supports voice calls, sending and receiving of SMS messages and General Packet Radio Service (GPRS), as well as the remote SIM card access, with usage of the Bluetooth connection, and the SAP [6] protocol. The device can be controlled remotely from a computer using AT commands ([3GP98c] and [3GP00]). It is done through serial connection with usage of the RS232 port.

The MS module uses extensively Converter-Multiplexer (CMUX) mode [7] of operation, for example to transmit data in the remote SIM card mode which is used very often in our project. The CMUX mode enables to transmit the different data, for maximum four client applications, through the one serial port connection with usage of the multiplexer. Figure 4.4 presents a work scenario of the CMUX mode of operation.

4. Description based on the document [Com08] available also under `http://www.telit.com/module/infopool/download.php?id=555` [Online, last check 14.01.2011]

5. Telit Communications S.p.A., `http://www.telit.com`, [Online, last check 14.01.2011]

6. Detailed SAP protocol description in Bluetooth specification [SIG08]

7. Description based on the documentation [Com09a] available also under `http://www.telit.com/module/infopool/download.php?id=616` [Online, last check 14.01.2011]

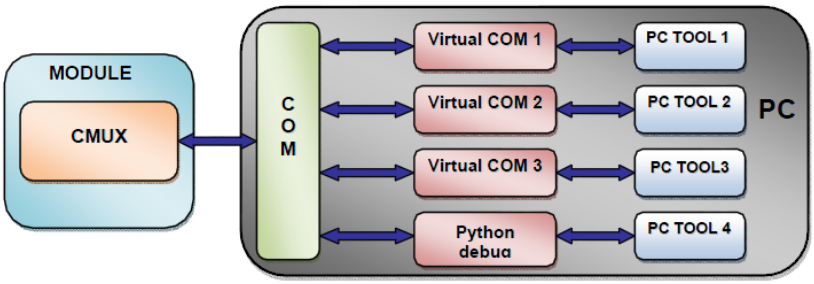

Figure 4.4: Telit GT864-PY CMUX mode of work example scenario from [Com09a]

The man-in-the-middle attack exploits an ability of the Telit GT864 module to communicate with the external SIM card with usage of the SAP protocol [8]. The protocol allows to access a data from the remote SIM card through the serial port or an additional hardware, for example by Bluetooth link. In this mode of operation communication occurs between a SAP client built in the Telit GT864 module and a SAP server. The SAP server provides information from a SIM card reader which allow the GT864 module to register in the GSM network. An additional side effect of this mode of operation is lack of time restrictions in accessing the SIM card – details in Chapter 6.1.1. The CMUX mode is activated through a serial port by the special AT command.

4.1.1.2 SAP server

It is possible to implement the SAP server, since the necessary documentation is published and available for everyone [SIG08]. This implementation process would be long; it also needs the time consuming debug phase to avoid new errors. Therefore we decided to use the already existing SAP server. For test purposes, the Samsung Corby [9] mobile phone was used, but any phone supporting the SAP protocol will be appropriate. The phone is presented in figure 4.5.

8. Description based on the documentation [Com09b] available also under `http://www.telit.com/module/infopool/download.php?id=576` [Online, last check 14.01.2011]

9. Detailed information about the Samsung Corby specification available at `http://www.cnet.com.au/samsung-s3650-corby_specs-339300123.htm` [Online, last check at 05.02.2011]

Figure 4.5: Samsung Corby from Samsung official website - `http://www.samsung.com`

The connection with the device is done through Bluetooth. Before establishing the connection a user should know the Bluetooth address of the SAP server.

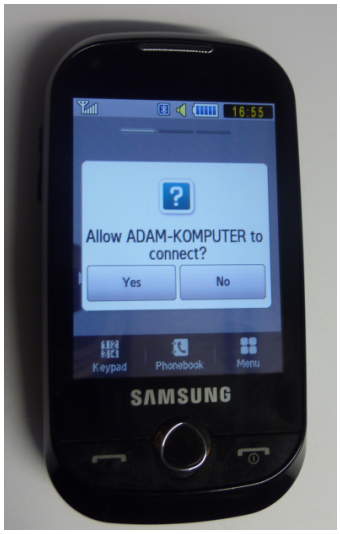

Figure 4.6: Query for the bluetooth's connection acceptance on the Samsung Corby phone

In a future version of the project, the SAP server from Samsung Corby may be substituted with a custom C++ implementation of the server, which receives the data from the smart card reader, or by a SIM card simulator. It may improve the speed of the whole hardware setup as described later in the Chapter 6.1.4. Before the connection is established, it is necessary to confirm the activation of the SAP server Bluetooth connection, as in figure 4.6.

4.1.2 Software

There are two main software components of the fake phone: drivers and the control script. The factor which determined the selection of the whole work environment was the availability of the CMUX drivers for Telit GT864-PY mobile station module. The CMUX command described in the [Com09a] allows the module's user to activate the remote SIM mode of operation. In this mode we may bypass SIM card access time restrictions as described in section 6.1.1 what is crucial for the whole project.The control script was written in Python version 2.5. This programming language is used and officially supported by the Telit company for all their products. As a consequence we may access and reuse official Telit company resources: scripts, the official product documentation and the official Telit GT864-PY Python libraries.

4.1.2.1 Drivers

The Telit company provided the CMUX drivers only for the Windows operating system environment. At the moment of a fake phone design there was no implementation of those drivers for any Linux operating system distribution. The creation of custom CMUX drivers for the Linux is possible, however it would extend significantly the necessary time for the project's development. In this version official drivers were used. Therefore, they require the Windows operating system as the fake phone basis. That led to division of the project into two modules while the fake BTS uses the OpenBTS software working only in the Linux environment.

It is necessary to install and run drivers before starting the script controlling the fake phone. Drivers necessary for the Bluetooth communication should be chosen according to the particular PC computer hardware. It also applies to drivers responsible for the serial port. On its own Telit GT864-PY module does not need any other dedicated software to operate.

4.1.2.2 Scripts

Main script controls all the equipment and synchronizes all of the operations in time. At the beginning it initializes the connection with the MS module through the RS232 port. Next it disables auto registration of the module to a GSM network. That will prevent the situation in which we will start the authentication before the end of the device's boot time, which as was proved during the tests, results in the rejection by MNO [10]. Later it initializes the Bluetooth connection with the SAP server in the mobile phone. After success of the previous action it switches on the device's CMUX mode of data transmission and activates the remote SIM mode of operation.

From that moment two additional threads are initialized. The main thread waits for the incoming transmission, from the MS module, through the serial port. In case of irrelevant for the attack commands it forwards them to the SAP server through the previously established Bluetooth connection. If commands which arrived are crucial for

10. More information in Section 6.1.2

the attack it has to set the appropriate semaphores for other two threads. In case of the command with RAND it additionally forwards the number to the fake BTS.

The first child thread, later in the text called thread one, is responsible for forwarding the responses from the SAP server to the MS module using the serial port socket. In case of responses which are crucial for the attack's success (belonging to the security or authentication commands) it modifies them with the data obtained from the fake BTS (IMSI, SRES, Kc).

The second thread, later in the text called thread two, controls the MS module. It waits 5 minutes in order to give MS time necessary to boot and later sends command to start the GSM network's registration process. Finally it waits for the information about the registration result (successful or not). In case of success, to create the proof of a attack, it may additionally order procedure of sending an SMS to the selected number on victim's expense.

4.1.3 Environment

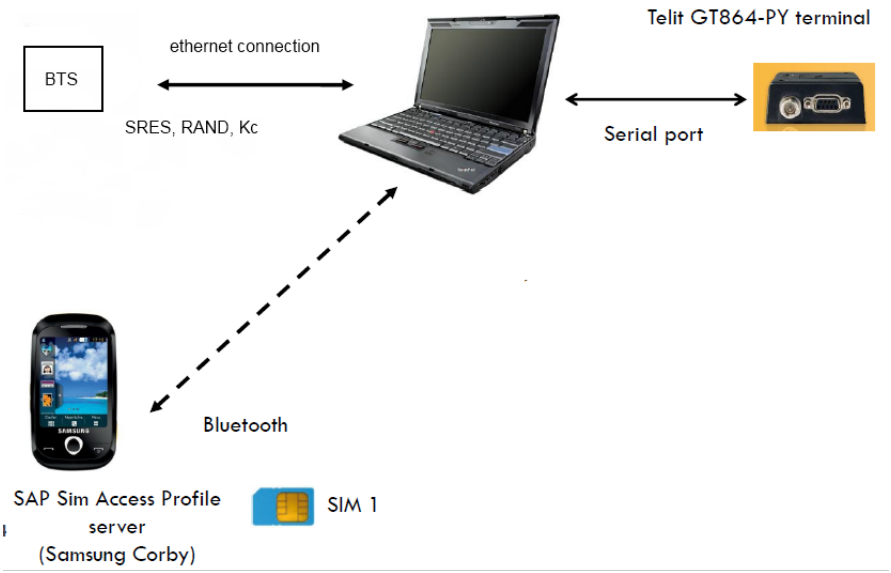

Figure 4.7: Fake phone – the final environment setup

Work environment is presented in figure 4.7. Workflow procedure begins with starting the main script, which waits for IMSI of the victim's MS. It will be send from fake BTS through the ethernet connection. During the next step our script connects to the terminal and enable the remote SIM card mode, what activates the connection with a SAP server in Samsung Corby. Later all SAP commands from the Telit module are forwarded to the SIM 1 card in the this phone. When the module asks about the IMSI, we modify the response with the IMSI obtained previously from the fake BTS over.

After that we have to wait 4 to 5 minutes for the boot procedure of the MS. Later we may order to begin the GSM network registration procedure and wait until the network will send RAND. Instead of passing it to the SIM 1 we send it to the fake BTS. In response we will receive the SRES and Kc from the victim's phone. We send them back to Telit module, in modified SAP command, and wait until we will receive information about the result of the registration. We may also send an SMS to prove the success.

During the project's test phase, we have used a smart card reader instead of the fake BTS. It allowed to obtain fast reliable results, which increased speed of work and created a solid base for further development. As written previously in this chapter Samsung Corby was used only to avoid implementation of the SAP server which would extend significantly project development time. The environment setup used in tests is preseted on Figure 4.8.

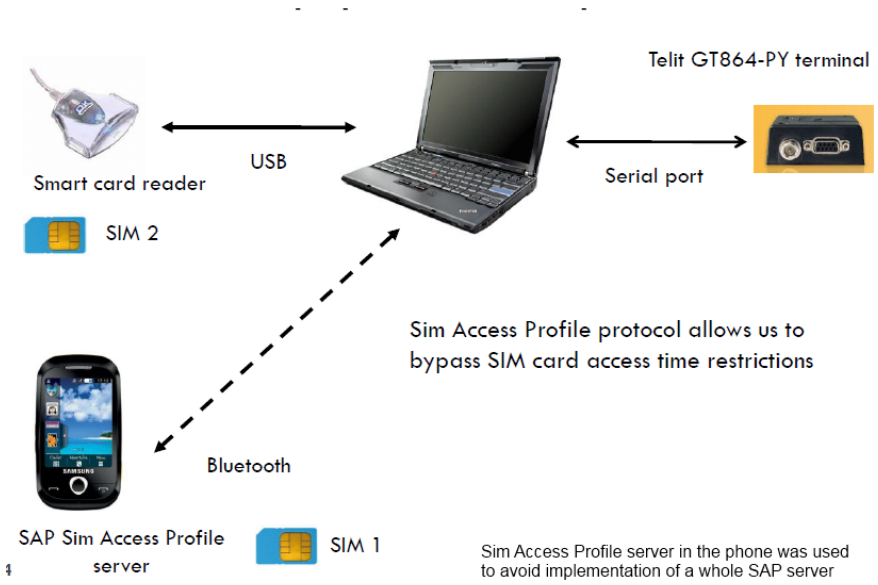

Figure 4.8: Fake phone – the test environment setup

4.2 Fake BTS

The fake BTS is the most complicated and advanced device used in the man in the middle attack. It should allow creation and running of the fake GSM network which can imitate the real MNO [11]. During the attack a victim's mobile station is lured to connect to this GSM network instead of the original one [12]. As a result the attacker can later obtain from the victim the authentication data (SRES and Kc). All actions connected

11. Detailed description in Section 2.4.2.1
12. Detailed description in Section 2.4.2.3 and 2.4.2.4

with this operation should be done in as short time as possible in order to decrease the delay in communication with a real MNO. The fake BTS should allow an exchange of the data with the fake phone, through a synchronized connection. In a situation when more than a one mobile station will be connected to the fake BTS the attacker should have a possibility to select the target, preferably by choosing the right IMSI number. The traffic from the victim's phone (e.g. SMS messages, voice calls) should be forwarded to the fake phone, what will reduce victim's chance to discover the attack. Finally the fake BTS together with the fake phone should enable multiple variants of malicious actions such as for example conversation eavesdropping, making calls and creating SMS messages on victim's expanse and many more [13].

The list of the fake BTS design goals is much longer than similar one in the case of a fake phone. They can be divided into three categories. In the first we will find all which are connected with a creation and running of the fake GSM network. It will need to have full capabilities of a real GSM network, for example voice transmission, SMS messages support, etc.

Second group will contain operations which are connected with breaking the GSM authentication and encryption. It includes obtaining of SRES and the Kc. In this thesis, in order to accomplish this goal an implementation of the known-plaintext attack on the GSM communication encrypted with cipher A5/2 was used. Detailed explanation one can find in section 2.5 and in [BBK07]. In the last group we have goals connected with communication between the fake phone and the fake BTS. In this thesis we assumed that the victim's mobile station is already connected to the fake GSM network, created by an attacker, or that the victim's mobile station can be forced to do it on the attacker's demand. This thesis does not solve problems connected with the victim's mobile station acquisition or luring.

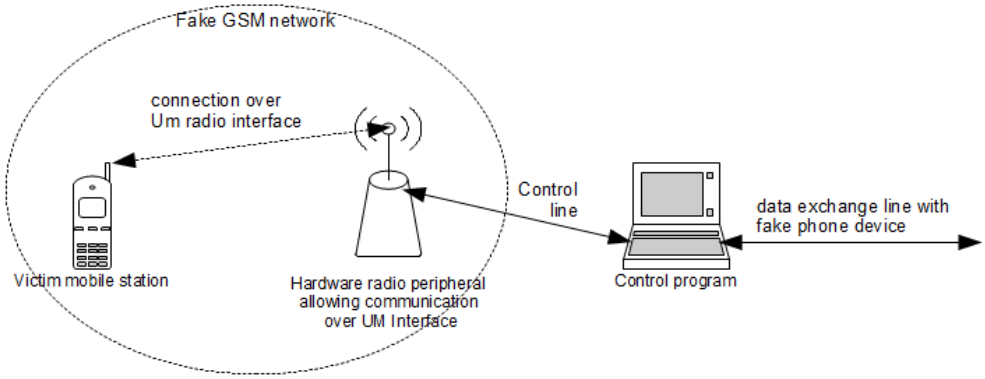

Figure 4.9: Fake BTS - the principle of work

13. Detailed description of the proposed attack scenarios in Section 6.4.

The theoretical diagram of the fake BTS implementation is presented on the Figure 4.9. Example workflow scenario will begin with creation of the owned by an attacker GSM network. Next step would be to lure a victim's MS to connect to this network, the described earlier victim acquisition operation. It may happen that more than a one mobile phone will be connected to the attacker fake GSM network. As a result attacker has to select the victim by selecting its IMSI. Later we have a process of data exchange with the mobile station and between the fake BTS and a fake phone. This communication is done in order to authenticate the fake phone as the victim's mobile station in the real GSM network. Final goal is to conduct at least one of the attack's scenarios. [14]

4.2.1 Hardware

In the practical fake BTS realization two main hardware components were used: radio device and PC.Radio device to create the radio Um interface,communication channel between a user's mobile station and a GSM network. [15]

PC class computer. It will support operation of the Um interface by processing part of the data from the radio peripheral. It will also control the radio peripheral over the USB interface. Finally, it will host the application responsible for the creation of the software-based GSM access point, which will implement the GSM stack. This will give a possibility to create the stand alone GSM network infrastructure. The last activity is the exchange of authentication data with the fake phone done through the ethernet connection.

The selection of the hardware components was determined by a decision to use the OpenBTS software as the GSM stack implementation. OpenBTS is described later in the Section 4.2.2.1. Its hardware requirements contain, between the others, the USRP peripheral device and computer capable of running Linux operating system with at least one USB interface. Full list of used hardware is described in the next section.

4.2.1.1 Equipment list

The main hardware component of the fake BTS is the regular stationary PC class computer equipped with the running Ubuntu distribution of the Linux operating system, as well as ethernet and USB controller. In order to create the Um interface the Universal Software Radio Peripheral (USRP) [16] was used. OpenBTS at this stage of development [17] does not support USRP 2. During tests several mobile phones were used. Most of the tests were conducted on the Nokia 6300. [18]More information about mobile phones supporting the A5/2 cipher is in the section 6.2. For our testing purposes we used also several SIM cards e.g. : Tschibo (virtual O2 network operator), O2 and

14. Detailed description of attack scenarios is available in section 6.4
15. Detailed description in the section 4.2.1
16. Brief description of the USRP device in the section 4.2.1.2
17. Current open OpenBTS release is OpenBTS 2.6 Mamou data from `http://openbts.sourceforge.net/` taken at 14.02.2011
18. More information about Nokia 6300 available at official product website available at `http://europe.nokia.com/find-products/devices/nokia-6300` [Online, last check 14.02.2011]

Vodafone. Detailed list of the equipment necessary to run the OpenBTS is described in the [Apv10].

Figure 4.10: USRP used in the project

4.2.1.2 USRP

The USRP stands for Universal Software Radio Peripheral, and the device used in the project is produced by the Ettus [19] company. It is the intermediate frequency section and digital baseband of a universal radio communication system. USRP in its work divides the operations between the host computer and the device itself, conducting all operations which need high computational speed. In order to achieve that a built in Field Programmable Gate Array (FPGA) is used. [20] FPGA also forwards row data through USB 2.0 controller to the host. Host computer performs all waveform-specific operations for example modulation and demodulation. In the project we used a modified version of USRP. It has two patched RFX1800 boards and an external clock. The external clock allows the device to operate in the frequency range important for GSM (GSM 900 MHz and GSM 1800 MHz). Thanks to that we may create the base station. In order to allow simultaneous transmission and reception of the data, two transceivers RFX1800 are installed. Detailed description in [LLC09]. In figure 4.10 we presented device used in the project, and in figure 4.11 we present the mainboard of the similarly modified USRP. We needed also simple antenna for the RFX1800 module.

19. Ettus Research LLC `http://www.ettus.com/` [Online, last check 15.01.2011]
20. Description based on the document [LLC09].

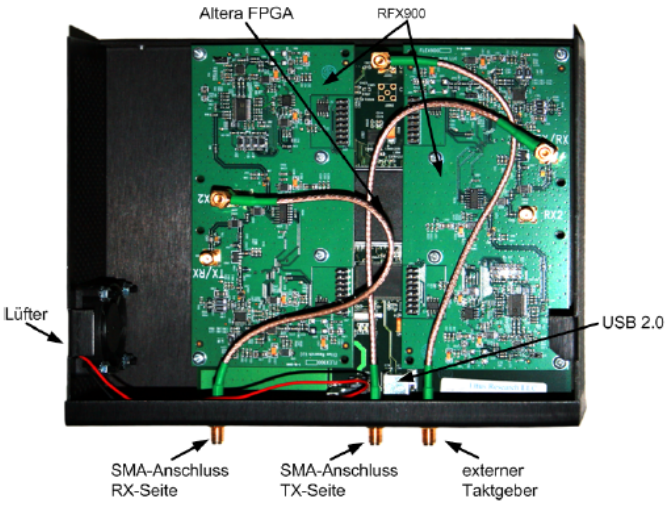

Figure 4.11: USRP motherboard with two RFX900 modules - figure taken from [Weh09]

4.2.2 Software

The main fake BTS software component is a set of applications allowing creation of the stand alone GSM network. They have to implement the GSM stack and give the possibility to conduct at least phone calls and SMS message exchange.

In order to achieve this goal we decided to use the OpenBTS software. This decision influenced significantly the design process of main project components. OpenBTS was written for a Linux environment (preferably a Debian distribution). In the project, the Ubuntu 10.10 Linux distribution was used as a basis of the fake BTS. Because of the fact that the fake phone, due to driver availability problems, works only in a Windows environment, there was a necessity to create a custom protocol which will allow data exchange through the ethernet network. Basically the whole environment contains a patched version of the OpenBTS, installed GNU Radio, and Asterisk server. All of the components are described later in this chapter.

OpenBTS is written fully in C++, and that is why all patches as well as new code parts were written in this programming language. All components are released under the GNU license which allows their free usage and decreases the cost of the project.

Additionally OpenBTS was modified by custom software patches allowing for example: breaking of the GSM A5/2 encryption, obtaining on demand the Kc security key, connection, synchronization and data exchange with the fake phone necessary to conduct the man-in-the-middle attack.

4.2.2.1 OpenBTS

OpenBTS [21] is a set of open source software that by replacing of the GSM network infrastructure (from the BTS upwards) allow radio-amateurs to create their own GSM networks. Calls are send to the Asterisk PBX server running on the same machine instead of a MSC. [22] Users of a system despite of calling each other may also send SMS messages. Such an OpenBTS based GSM network consists of a USRP, connected to a USB port of a computer with Linux, Asterisk, GNU Radio and OpenBTS. The structure of the network is presented on the Figure 4.12. OpenBTS is written in the C++ programming language and released as free software under the AGPL license. OpenBTS's custom modifications may allow an attacker to break the A5/2 cipher and introduce communication with the fake phone. It served in the project as basis for fake BTS implementation.

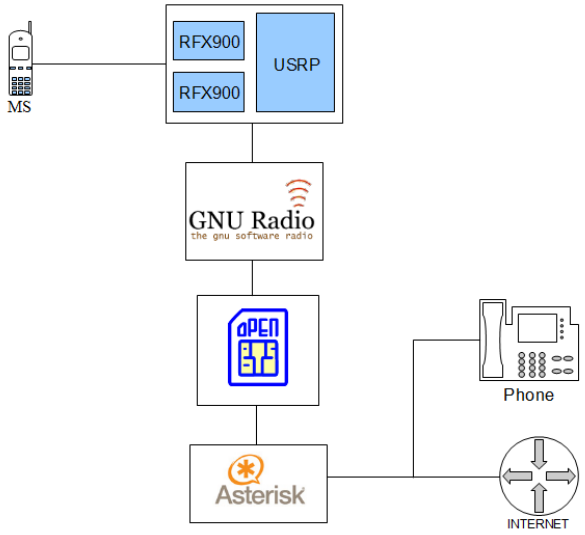

Figure 4.12: Structure of the OpenBTS

The mobile station connects first to the USRP. This is the only hardware module which enables a radio communication. The transmitted data are later passed to the GNU Radio software. The GNU Radio serves for the OpenBTS as a USRP interface (it can operate the USRP only through the GNU Radio). The USRP, GNU Radio and OpenBTS create together the Um interface. Finally the Asterisk server was used to do a user management and call forwarding, it fulfills the functions of HLR and AuC. A detailed presentation of the environment is included in figure 4.13.

21. OpenBTS project `http://openbts.sourceforge.net/` [Online, last check 15.01.2011]
22. Description based on [Apv10]

In the OpenBTS used in the man-in-the-middle attack the following modifications have been introduced: *a*) Patches designed by Dipl.-Ing. Janis Danisevskis which enable authentication and encrypted connections. They introduce the `MM Authentication Request` message, `RR Authentication Response` message, `RR Ciphering Mode Command` message, `RR Ciphering Mode Complete` message as well as usage of A5/2 dynamic libraries. These patches, created originally for OpenBTS 2.5.4 Lacassine version, were ported and modified by us, as a part of the thesis, to the newer version of the project – OpenBTS 2.6.0 Mamou. *b*) Patches designed by Dipl.-Ing. Janis Danisevskis which allow to break the A5/2 cipher and get the Kc key. Patches created originally for the OpenBTS 2.5.4 Lacassine version were ported and modified by us, as a part of the thesis, to the newer version of the project - OpenBT 2.6.0 Mamou. *c*) Finally the patch which enables the man-in-the-middle attack over the Um interface. It adds a new command "attack" to the OpenBTS command line interface. It enables communication with the fake phone and converts each of the mobile stations connected to the running OpenBTS into the oracle which on administrator request returns a SRES and a Kc for a given RAND.

4.2.2.2 GNU Radio

The GNU Software Radio [23] (originally Software Defined Radio) is a software emulator of the hardware which allows processing of the high frequency transmitter or receiver signals. It is used mainly as the USRP controlling program which prepares data for signal processing. The signal processing operation is later done by custom modules. GNU Radio offers an easily reconfigurable radio system which allows its users to create different devices without the need to buy several expensive radios.

4.2.2.3 Asterisk

Asterisk [24] is a software server which allows users to conduct phone calls between each other, call public phone numbers and conduct Voice over Internet Protocol (VoIP) communication. Basically it has possibilities of the telephone private branch exchange. It has a dual license, GNU and proprietary software license permitting to distribute closed source components. At the beginning Asterisk was designed for the Linux operating system. In this project it works under the Ubuntu Linux distribution. From the wide spectrum of possible Asterisk's features (such as voice mail, conference calling, interactive voice response) in the project we are using mainly only the basic ones, such as phone calls and user management. However, the broad variety of them may be very useful in the further project development.

23. Information in this chapter are based on `http://gnuradio.org/` [Online, last check 28.02.2011] and [Weh09]

24. Information in this Chapter is based on the `http://www.asterisk.org/` original site [Online, last check 28.02.2011]

4.2.3 Environment

In figure 4.14 we presented placement of hardware and software components in the final fake BTS environment.

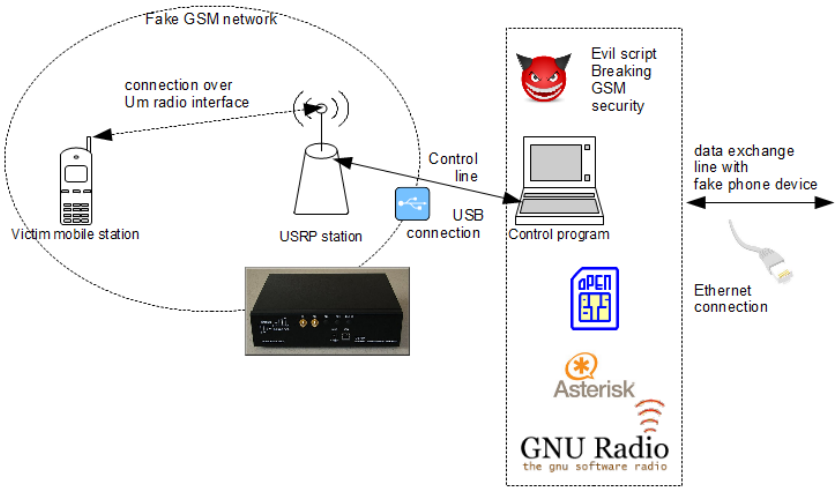

Figure 4.14: Fake BTS diagram with hardware and software components placement

Final workflow procedure begins when victim connects to the fake BTS network over the radio Um interface. The attacker using the command line interface of the OpenBTS types attack command with IMSI number of the victim's mobile station which is forwarded to the fake phone through the ethernet connection. In the next step fake BTS waits for a RAND number from the fake phone. When the RAND number arrives, it starts a paging procedure in order to establish the connection channel with the victim's mobile phone. At the moment when the victim's mobile phone reports to the BTS the Standalone Dedicated Control Channel (SDCCH) channel is established. Later BTS sends to the victim's mobile station the `MM Authentication Request` message containing the obtained RAND number and waits until the `RR Ciphering Mode Command` message containing the SRES. BTS sends the SRES to the fake phone over the ethernet connection. The attacker's next step is to start by a BTS the encrypted connection by sending the `RR Ciphering Mode Command` message to the victim's mobile phone. In response the `RR Ciphering Mode Complete` message will arrive. In parallel, using the appropriate patch using dynamic libraries, it breaks the A5/2 cipher and obtains the Kc key. As soon as it is done Kc is sent to the fake phone. After that fake BTS can finish the connection and release the SDCCH channel.

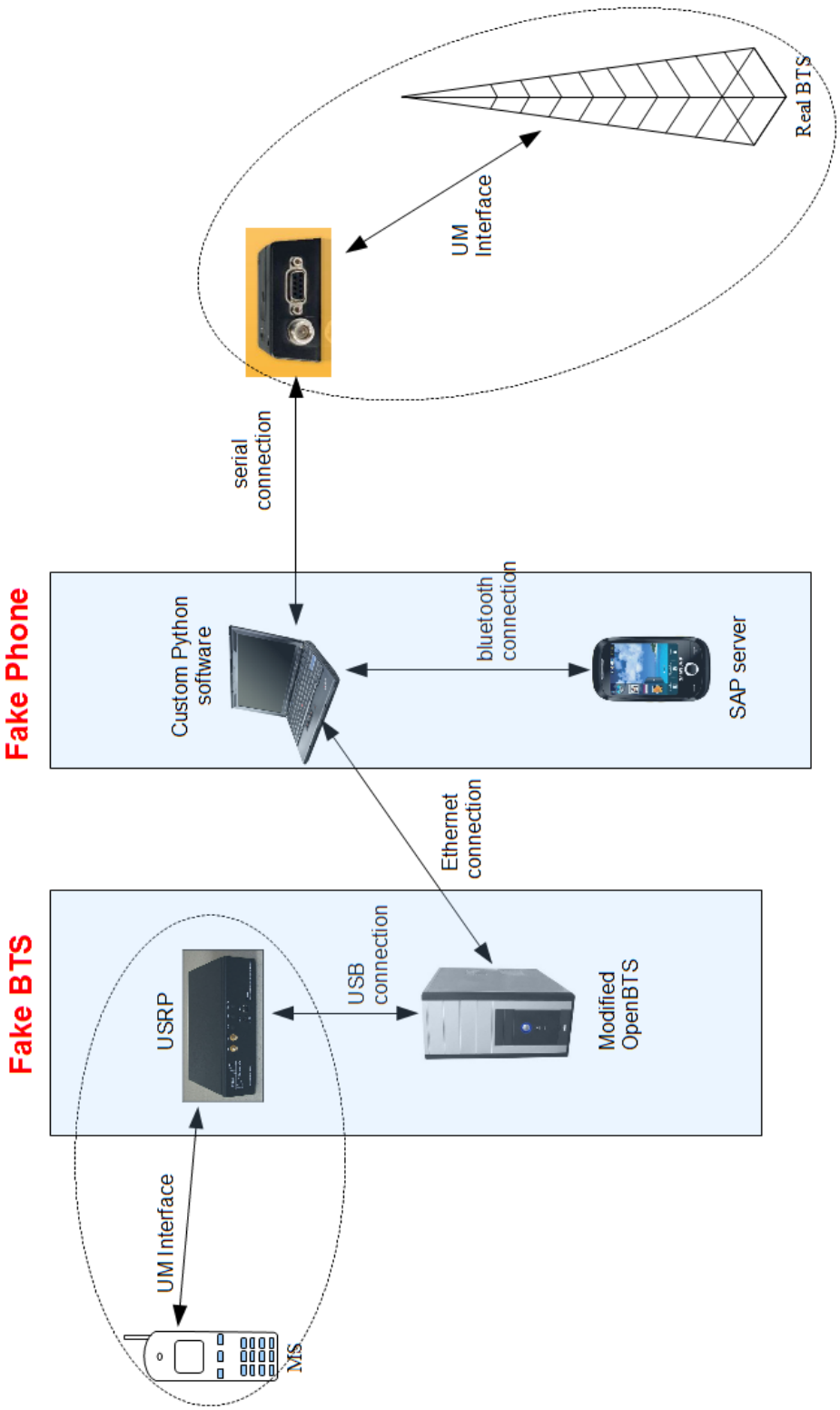

Figure 4.1: Equipment configuration in the man-in-the-middle attack

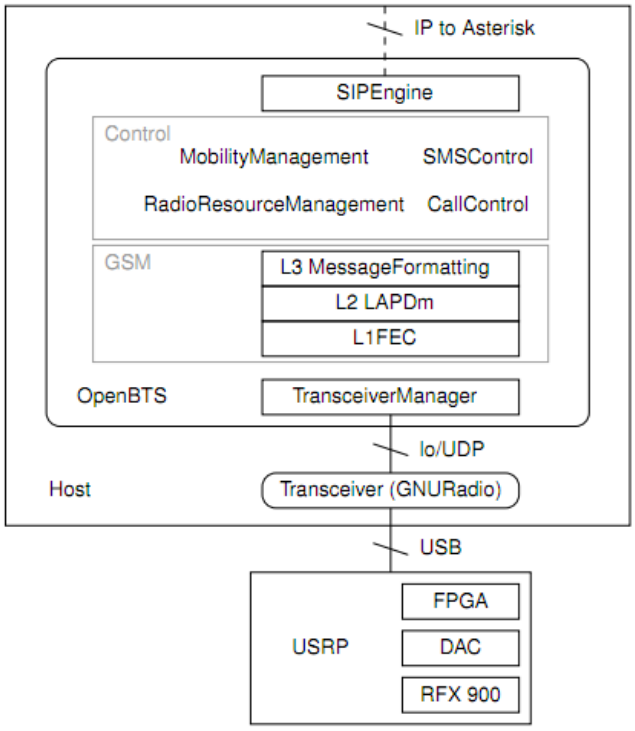

Figure 4.13: OpenBTS environment schematics. Figure created by Janis Danisevskis and included with his kind permission.

5 Implementation

In the previous chapter we described the design of two devices needed to conduct the attack: a fake phone and a fake BTS. The goal of the fake phone is to imitate a victim's mobile station to a real GSM network. The goal of the fake BTS is to imitate a real GSM network to the victim's mobile station. Both devices should be synchronized with each other and should allow forwarding of the GSM transmission with, preferably, no delay.

The next step is to conduct the test attack using those devices.The attack starts in the fake BTS station and is conducted in the following phases: *a*) acquiring the IMSI number of the victim's mobile station, *b*) booting the fake phone with the acquired victim's IMSI, *c*) obtaining a RAND number – starting the registration of the fake phone in the real GSM network, *d*) obtaining the SRES and the Kc from the victim's mobile station – based on the previously obtained RAND, *e*) finishing the fake phone registration with the SRES and the Kc.

This chapter presents details description of each of them. It finish with success. We were able to register our fake phone with stolen from victim credentials and send SMS on the victim expense.

5.1 Acquiring the IMSI of a victim's mobile station

As described in the Chapter 2.4.2, at the beginning the victim's mobile station is lured to connect to the fake BTS. Victim's mobile station starts registration to the fake network with the IMSI Attach procedure ([3GP98b]) [1]. First the channel establishment procedure is executed by sending the `RR Channel Request` *(CHAN_REQ)* message ([3GP98b] 9.1.8) on the Random Access Channel (RACH) channel and receiving response - the `Immediate Assignment` *(IMM_ASS_CMD)* message ([3GP98b] Section 9.1.18) on the Access Grant Channel (AGCH) channel. After that the mobile station switches to the assigned SDCCH channel and sends the `Location Update Request` *(LOC_UPD_REQ)* to the BTS. This message contains the IMSI code of the victim's mobile phone. The first goal of the attacker is already accomplished – the fake network receives the victim's IMSI.

In the OpenBTS the attacker may see all the phones registered to the fake network by displaying the list of TIMSI. It is available by typing the `tmsis` command. Result is presented in figure 5.1.

1. All GSM xx.xx references reefer to series of GSM specifications provided by 3rd Generation Partnership Project (3GPP) and available at http://www.3gpp.org/ **??**

```
OpenBTS> tmsis
TMSI       IMSI            IMEI              age  used
0x4ce547c0 208123456789012              ?  113h   84s
0x4ce547c1 208111111111111              ?  113h  113h

2 TMSIs in table
```

Figure 5.1: Result of the OpenBTS timsi command

5.2 Booting a fake phone with the acquired victim IMSI

The attacker starts the fake phone python script that is located on a machine connected with the fake BTS through the ethernet network. The script starts the boot procedure of the Telit GT864-PY [2] mobile station.

First it opens the serial port which the Telit GT864-PY station is connected to and with the usage of the `AT+CMUX=0,02` [3] command invokes the CMUX [4] mode of work of the device. After that the script disconnects from the port and reconnects once again to the first virtual serial port. It sends the `AT+COPS=2` [5] command through this port. The goal of this command is to switch off an automatic attempt of the mobile station to connect to the GSM network when connection with the SAP server is established. After positive response from the Telit GT864-PY, the mobile station script disconnects from the first virtual point.

Next the connection with the SAP [6] server is opened through the Bluetooth interface. After that the ethernet server socket is invoked and the script switches to the busy waiting mode of work. It is waiting for an IMSI code of the victim's mobile station. In order to start the attack the fake BTS owner has to use the command attack. The `attack` command takes the victim's IMSI number (which has to be valid) as a parameter.
For example `attack 262075002943081` will start the attack on the victim's mobile station with the IMSI `262075002943081`. The result is presented in figure 5.2.

The fake BTS starts the ethernet client socket and connects to the fake phone server. After the connection is established it sends the victim's IMSI and switches into the busy waiting mode. It waits for the RAND number from the fake phone. The reception of the victim's IMSI finishes the fake phone script's busy waiting loop.

From this moment, the script operates with the usage of three threads. Main Thread first connects to the second virtual port. It reads data from the Telit GT864-PY and forwards them to the SAP server through the Bluetooth interface. Thread One by using busy waiting reads a response from the Bluetooth socket. Later it forwards the response to the virtual serial port number two. Thread Two is connected to the first virtual port and controls the work of the Telit GT864-PY device. The main thread and the thread

2. Detailed device description in [Com08]
3. Command detailed description in [Com09a]
4. Detailed description in [Com04]
5. Command detailed description in [GT864MANUAL]
6. Detailed description in [Com09b]

```
OpenBTS> attack 262075002943081
Man-in-the-middle attack over UM Interface
Adam Kostrzewa 2011 TU Berlin
Target IMSI262075002943081
SOCKET :: send victim IMSI
waiting for RAND
SOCKET :: RAND received
RAND :: 91E322BBF8EA7F57957C2C3C5035D5E5
request send

OpenBTS> OpenBTS>  HERE
 SRES=0xe75d21e0
OpenBTS>  Kc=f8e2fd5f5e389000
```

Figure 5.2: Fake BTS attack command result

number one are responsible for the communication between the SAP server and the Telit GT864-PY module. The thread number two controls the fake phone.

The second thread sends the `AT#RSEN=1,1,0,2,0`[7] command, which activates the remote SIM card connection with the usage of the SAP protocol. Data are exchanged between the SAP server and the client until the IMSI request command[8] arrives which is detected by the main thread. It is done by reading of the EFIMS command with the '6F07' code. The main thread sets a flag for the thread number one. From this moment the first thread starts to wait for the IMSI response from the SAP server. When the response arrives it substitutes the IMSI from the SAP server with the victim's mobile station IMSI obtained from the fake BTS.

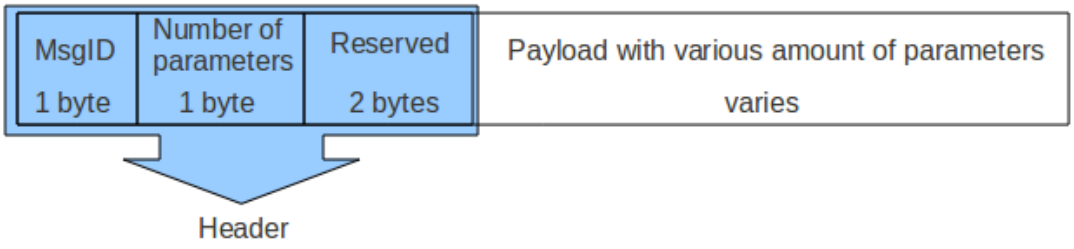

Figure 5.3: Structure of the SAP command.

A sample structure of the command coded with the SAP protocol is presented in figure 5.3 and structure of SAP command's parameter in figure 5.4.

7. Command detailed description in [Com09b]
8. Detailed description in [rGPPGT99] ch. 10.3.2 p. 55

Figure 5.4: Structure of the SAP command parameter.

After the successful IMSI substitution the script waits some time for the Telit GT864-PY station to finish the boot procedure. During this time the module reads data from the SIM card. Boot time of the Telit GT864-PY lasts usually from 4 to 5 minutes. In the test version of the script this time is set to 5 minutes. It was set basing on the results of the test.

According to the official documentation [Com09b] the beginning of the registration of the Telit GT864-PY module to a GSM network may happen at any time after the `Remote SIM Enable (RSEN)` command has been send. Tests proved however that if the registration is started in time shorter than 5 minutes the authentication request is always rejected by the network. Due to the usage of Bluetooth's and serial port's connections between the SAP server and the client communication, it may happen that this time extends. The heavy link usage and limited processing abilities of the device may lead to timeouts in the communication with the GSM network. Response to the authentication request is sent too late and therefore it is not accepted. The possible way to improve the implementation is to perform modification of responses from the Telit GT864-PY in a way that will guarantee the exchange of the data necessary to boot procedures. In the following way we may get rid of, for example, checking contacts stored in the SIM card. More attack's speed improvements propositions are presented in the Chapter 6.1.4.

Ability to start and successfully perform the GSM authentication procedure of the Telit GT864-PY mobile station with the victim's mobile station IMSI indicates end of this stage of attack.

5.3 Obtaining a RAND number

After a successful boot procedure the device may start registration to the network as described in the Chapter 2.3.2 . The registration is started in the thread number two by sending the `AT+COPS=0` [9] command to the virtual port number one. The Um interface authentication procedure is defined in detail in [3GP98b] standard Section 4.3.2 and [3GP98a] Section 3.3.1. Firstly the GSM network sends a 128 bit RAND number to the mobile station in the `MM Authentication Request` message. Later the RAND is passed to the SAP server in the `RUN GSM ALGORITHM` [10] command (identified by the number `'A088000010'`). Example of the `RUN GSM ALGORITHM` is presented in figure 5.5.

9. Command detailed description in [Com04]
10. Detailed description in [rGPPGT99] ch. 9.2.16 p. 46

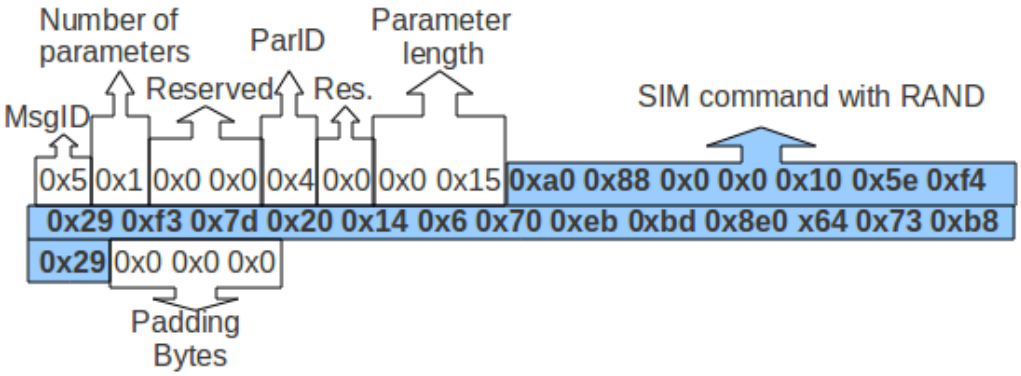

Figure 5.5: Example of RUN GSM ALGORITHM command structure.

The main thread of the fake phone script forwards this RAND through the ethernet connection to the fake BTS. It also sets up a flag for the thread number one in order to listen for the `RUN GSM ALGORITHM` command's response containing SRES and Kc numbers calculated by the victim's SIM card.

5.4 Obtaining SRES and Kc from victim's mobile station

The fake BTS finishes the busy waiting. It starts a paging procedure in order to establish the communication link with the victim's mobile station. In order to do that the fake network sends the `RR Paging Request` message ([3GP98b] Sections 9.1.22-9.1.23) over the Paging Channel (PCH), using the subscriber's IMSI or TIMSI as an address. The GSM standard does not allow paging by the IMEI ([3GP98b] Section 10.5.1.4). A successful response from the mobile phone leads to creation of the SDCCH channel.

Later with the usage of that channel the fake BTS sends the `MM Authentication Request` message to victim's phone. The fake phone responds with the `RR Authentication Response` message containing the SRES number. The SRES number is sent back to the fake phone. Next step is to obtain the Kc number. It is done by exploiting weaknesses of the GSM A5/2 encryption algorithm as described in [BBK07]. This procedure is initiated with the `RR Ciphering Mode Command` message indicating that the A5/2 mode of GSM cipher should be used. Parallel to that the Kc is recovered with the usage of decryption libraries written by Dipl.-Ing Janis Danisevskis. The process of enabling encryption should be finished by the reception of the `RR Ciphering Mode Complete` message ciphered with the usage of Kc. After obtaining Kc the attacker forwards it to the fake phone.

5.5 Finishing the fake phone registration with stolen SRES and Kc

The fake phone's thread number one receives SRES and Kc. Using them it modifies the `RUN GSM ALGORITHM` command response by the substitution its payload and sends it back to the GSM network.

Starting from that moment fake phone's thread number two starts querying the Telit GT864-PY about the registration status with the command `AT+CREG?` [11].

```
Starting the system...
1297877477.8922 FORCE 3075790608 Logger.cpp:109:gSetLogFile: setting log path
to test.out
1297877477.9192 ALARM 3075790608 OpenBTS.cpp:130:main: OpenBTS starting,
ver 2.6.0Mamou build date Feb 11 2011
1297877477.9696 FORCE 3075540688 Logger.cpp:194:gLogInit: Setting initial global
logging level to NOTICE
1297877477.9698 FORCE 3075540688 Logger.cpp:109:gSetLogFile: setting log path
to test.TRX.out
1297877483.508424 3075790608:

Welcome to OpenBTS. Type "help" to see available commands.

OpenBTS> attack 262075002943081
OpenBTS> attack 262075002943081
Man-in-the-middle attack over UM Interface
Adam Kostrzewa 2011 TU Berlin
Target IMSI262075002943081
SOCKET :: send victim IMSI
waiting for RAND
SOCKET :: RAND received
RAND :: 91E322BBF8EA7F57957C2C3C5035D5E5
request send

OpenBTS> SRES=0xe75d21e0
OpenBTS> Kc=f8e2fd5f5e389000
```

Table 5.1: Fake BTS console output

The registration's status changed to 1 (boolean value true) means that the registration was successful and right now the Telit GT864-PY mobile station impersonates victim in the real GSM network. In order to prove the success of the attack during tests an

11. Detailed information in [Com04] ch.2.4.7 p.25

SMS was send from the Telit module. That fulfills first attack scenario requirements – stealing identity by the dynamic SIM card cloning. The list of other possible attack scenarios is available in the Chapter 6.4. Log of the fake phone's console output and the fake BTS console output are presented in Table 5.2 and Table 5.1. In figure 5.6 it is presented the chart of the whole man-in-the-middle attack on the fake phone side. Figure 5.7 presents the chart of whole attack on the fake BTS side.

```
Microsoft Windows [Wersja 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzezone.

C:\Users\Adam>cd Desktop\magisterka\python_scripts
C:\Users\Adam\Desktop\magisterka\python_scripts>03.py
COM4 -> opened
GT864 terminal -> enabled CMUX mode of work
COM4 -> closed
COM7 -> opened
GT864 terminal -> disabled registration to GSM network
bluetooth -> searching all nearby bluetooth devices for SAP service
bluetooth -> connecting to "Sim Access" on A8:F2:74:09:A6:14
bluetooth -> connected


SOCKET -> waiting for IMSI .....
SOCKET -> client is at ('10.0.10.86', 37482)
SOCKET -> rcv IMSI: 262075002943081
bluetooth -> waiting for SAP Server start ...
bluetooth -> waiting for SAP Server start ...
bluetooth -> waiting for SAP Server start ...
bluetooth -> waiting for SAP Server start ...
bluetooth -> waiting for SAP Server start ...
bluetooth -> SAP Server is running
GT864 terminal -> waiting for SIM data load procedure to finnish (5 minutes)
SAP Client -> IMSI Request
SAP server -> IMSI ativated
SAP server -> IMSI ativated
SAP server -> IMSI ativated
SAP server -> new IMSI is 262075002943081
GT864 terminal -> starting fake phone registration to GSM network
SAP Client -> RUN GSM ALGORITHM command
SAP server -> cipher activated
GT864 terminal -> waiting for registration....
SAP server -> cipher activated
4.11679908021 seconds process time
SAP server -> new KC (RAND SRES Kc) are :
91E322BBF8EA7F57957C2C3C5035D5E5 E75D21E0 F8E2FD5F5E389000
GT864 terminal -> waiting for registration....
GT864 terminal -> fake phone is registered to GSM network
```

Table 5.2: Fake phone console output

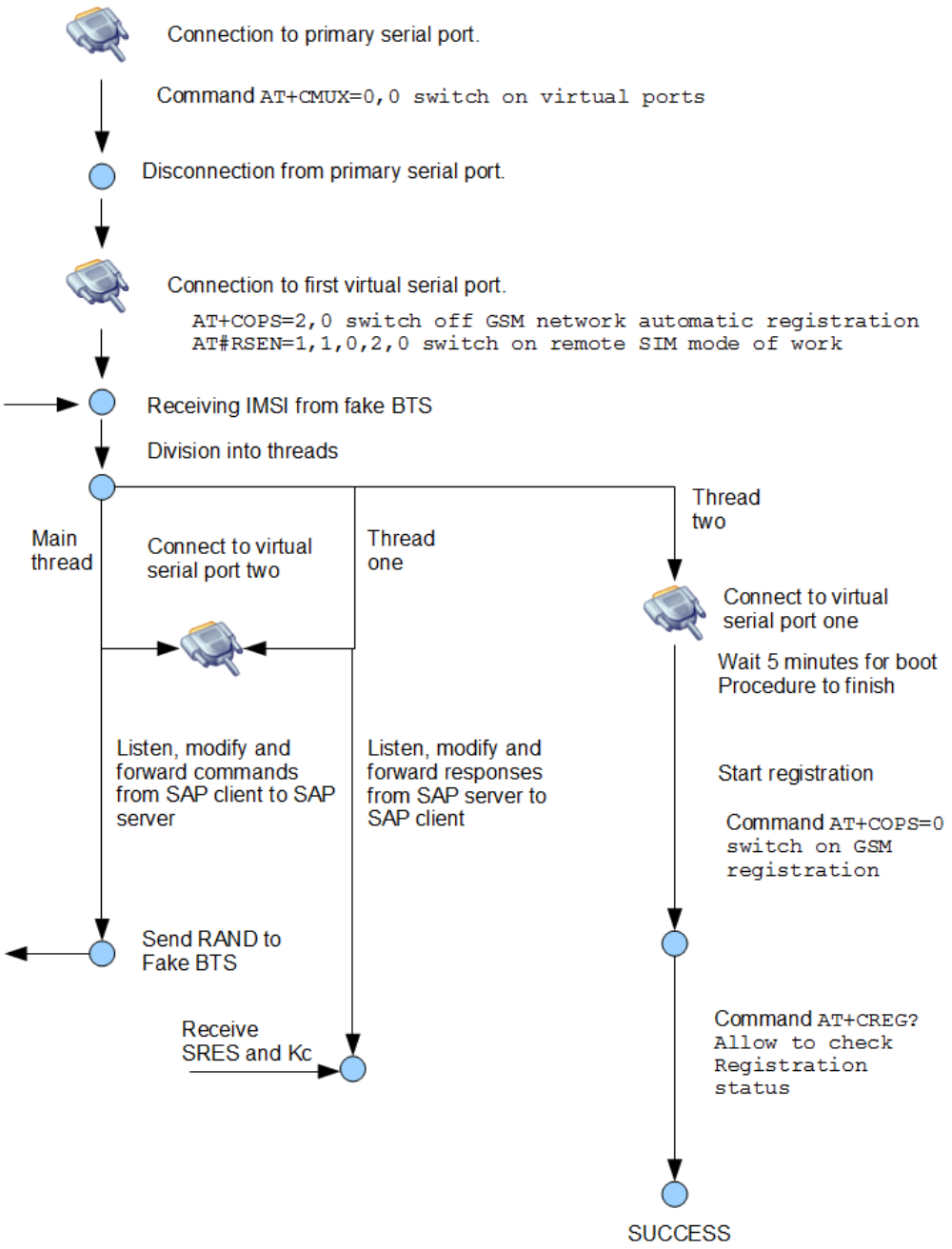

Figure 5.6: Fake phone practical workflow diagram

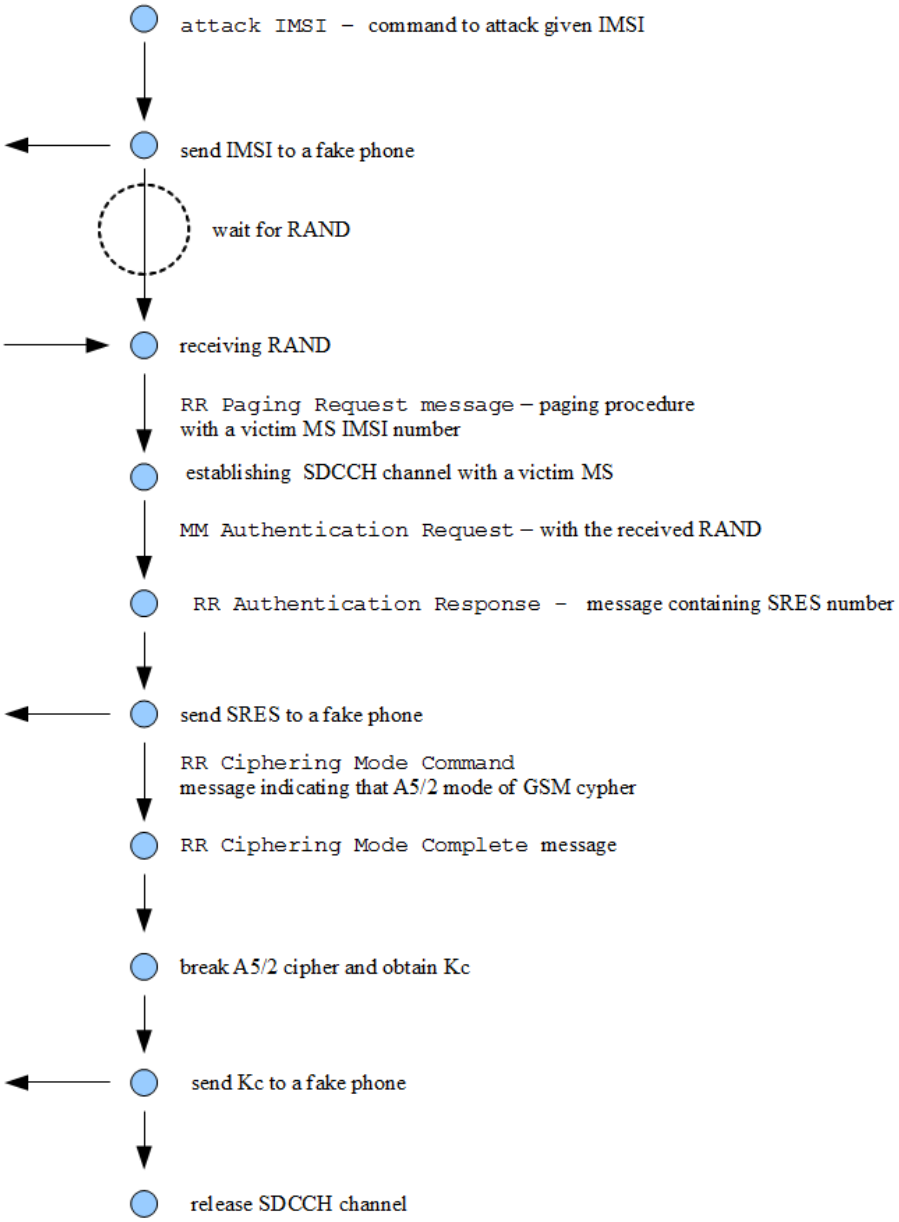

Figure 5.7: Fake BTS practical workflow diagram

6 Evaluation

In this chapter we would like to present the evaluation of the equipment created in the project. It will be based on the research and tests done by us. We would like to assess what impact the man in the middle attack can have in practice and what are possible methods to prevent it or at least decrease its consequences. We will start from presentation of attack's limiting factors such as time restrictions which appeared to be crucial for its success. They are the most important limiting factor and their modification may increase significantly attack's complexity.

Second problem considered by us is the hardware support of the A5/2 cipher by manufacturers in their devices. Without this support there not exist any known attack based on A5/2, according to the present scientific knowledge and research .

Next we would like to present the approximate cost of attack. The cheaper it is the bigger threat it present. The reason for this is that low cost subsequently increase the amount of people which are capable to afford attack. That increases its probability and strength while with increased popularity new ideas and improvements will arise.

Last section present the possible attack scenarios. We would like to show the designed equipment capabilities and usage options. The more of them exist the more dangerous attack can be in practice and more variants it can have.

6.1 Timing aspects of the attack

In figure 6.1 we presented the flow chart with states which are important for the time evaluation of the attack. We will frequently refer to them later in this section.

Tests showed that the crucial factor for a success of the attack is the time necessary to send back the response with a substituted authentication data. This time is counted from the moment of receiving theRAND number from the GSM network (state D) to the moment of sending back the authentication response (state E). When this action takes too long, we may get the timeout and the registration will be rejected by MNO. In case of a regular mobile station with a build-in SIM card reader the most important factors influencing this time are: the distance from the BTS station to the mobile station and possible transmission's distortions. The time necessary to obtain SRES and Kc from the subscriber's SIM card is in the order of milliseconds and does not influence the operation result.

In case of the man in the middle attack we are using the much more complicated equipment setup. As a consequence additional factors appear which may extend significantly time necessary to send back the authentication response [1]. The most important

1. Detailed description of the attack implementation can be found in Chapter 5

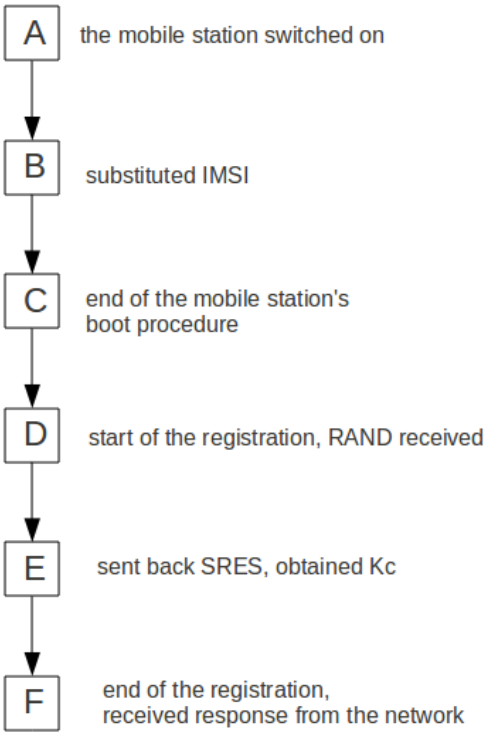

Figure 6.1: Man-in-the-middle attack time flow chart.

of them are: *a*) technology in which communication with the SAP server is done - Bluetooth connection, *b*) time necessary for the fake phone to do a checkup of messages in order to find the appropriate command to substitute (transition from the state A to B and from D to E), *c*) time necessary to communicate with the fake BTS over the ethernet connection (transition from the state D to E), *d*) time necessary for a fake network to conduct the paging procedure in order to create the communication channel with a victim's mobile phone (transition from the state D to E), *e*) time necessary to send the `MM Authentication Request` message and receive `RR Authentication Response` with SRES number (transition from the state D to E), *f*) time necessary to conduct the instant known-plaintext cryptanalysis of the GSM encrypted communication in order to recover Kc (transition from the state D to E), *g*) time necessary to parse commands and do the substitutions (transition from the state A to B and from D to E).

Methods used to decrease these delays as well as precise time measurements are discussed later in this section.

6.1.1 SIM card reader response time

Time of getting response from the SIM card is very short – in the order of milliseconds. In case of regular SIM card's readers this time is additionally restricted to a very small

values. Violation of those restrictions leads to the rejection of the response from a SIM card directly by MS, without sending it to the network. In the man in the middle attack there is a need to substitute the values from the attacker's SIM card response with the data obtained from victim's mobile station through the radio interface. This process may be quite long. Definitely it exceeds all possible restrictions which are valid for regular SIM card's readers.

We discovered that there exists a possibility to bypass those limitations. The Telit GT864-PY mobile station working in the remote SIM mode does not have the restrictions described earlier. For receiving the response data from the SAP server the busy waiting mechanism is used. This mechanism allows an attacker to bypass the SIM card access time restrictions.

In order to test the Telit GT864-PY reaction for an extended SAP server response time we connected it to a PC with an active Bluetooth link to a SAP server in the Samsung Corby phone. Later we waited for the IMSI request command send from the Telit module SAP client in order to forward it to Samsung's phone and wait again for the response from its SIM card. When the command arrived we modified it as in the case of regular attack. Later we waited the appropriate artificially introduced delay time before sending it back. The final test action was to check if the device is responding. In case of a failure it should hang up.

Delay time (sec)	2	5	10	20	40	60
Response acceptance	Success					

Table 6.1: Delay time in response from the SAP server and results its acceptance by Telit GT864-PY mobile station

As presented in the Table 6.1, for times shorter than 60 seconds no rejection was encountered. Taking into consideration that according to tests and measurements the average registration time during attack is 4.7 second [2], this time is more than enough for the attacker's purposes.

6.1.2 Telit GT864-PY boot time

During the development phase of the project we observed that it is impossible to obtain the instant Telit GT864-PY module registration to a GSM network in the remote SIM mode of work. According to the Telit manual [Com04] registration can be started at any moment of the device work by sending the `AT+COPS=0` command through the serial port. When device is in the internal SIM card mode of work, in which a SIM card is inserted to the smart card reader build in it, it behaves as it was described in the documentation. Experiments showed that if we use the data received from an external SIM than submitting the `AT+COPS=0` command will lead to rejection of the registration by a GSM network. Network will send later four RAND numbers and the device will switch off registration attempts for some time. However if we will wait appropriate

2. Detailed information in Section 6.1.3

Time (sec)	Result
Immediately	Failed
100	Failed
150	Failed
180	Failed
200	Failed
210	Failed
220	Success
240	Success
300	Success
340	Success

Table 6.2: Telit GT864-PY mobile station boot time necessary for successful registration to the GSM network

amount of time, later in this section called boot time, and than issue the command we result in success.

In order to estimate the length of the boot time we conducted experiment in which we were submitting the `AT+COPS=0` command (state D) with a gradually increasing delay time, counted from the moment of starting the device (state A). Results are presented in the Table 6.2.

There can several possible explanations of this device's behavior. For sure due to the usage of the Bluetooth link and the serial port connection between the SAP server and the SAP client, the time necessary to do the communication rises. In our opinion probable explanation can be that this time delay, in case of heavy link usage connected with limited processing abilities of the Telit GT864-PY causes timeouts in the command processing by the mobile station and results in distortions in the communication with the GSM network. As a result the response to the authentication request is sent too late and therefore gets rejected.

6.1.3 GSM network access time

GSM network's access time is the time which flows from the moment of receiving the `Authentication Request` command (containing RAND) from the GSM network (state D) to the moment of sending the `RUN GSM ALGORITHM` command response (containing SRES and Kc) using SAP protocol to a mobile station (state E).

In the first version of the environment the authentication response data, Kc and SRES, were sent after the authentication and encryption procedure finished. That gave overall GSM network's response time higher than 12 seconds (transition from the state D to F). That was unacceptable and resulted in rejection by MNO. In order to decrease this time we send the Kc and the SRES to the fake phone independently, as soon as they are received (or calculated), without waiting for the finish of the encryption procedure. This allowed to reduce overall time to the average of 4,7 second (transition from the

Overall delay time(sec)	Network	
	o2	Vodafone
10,4	Failure	Failure
9,42	Failure	Failure
8,4	Failure	Failure
7,82	Failure	Failure
7,54	Success	Failure
7,47	Success	Success
5,40	Success	Success
4,39	Success	Success
3,48	Success	Success
1,47	Success	Success

Table 6.3: GSM registration time delay

state D to F). That appeared to be sufficient to obtain successful registration during each attempt.

In order to check how much time can the attacker possibly have, to conduct all his actions, we designed an experiment. We used Telit GT864-PY module and smart card reader with a SIM card inside. We set up Telit device in the remote SIM card mode connected to the PC with active Bluetooth connection to an SAP server in the Samsung Corby mobile phone and connection to the smart card reader. Later we done the following steps: *a*) wait for the `MM Authentication Request` message from the network, *b*) forward a RAND to the smart card reader, *c*) wait for the response, *d*) substitute the Kc and the SRES into the response of the `RUN GSM ALGORITHM` command,as in case of regular attack, *e*) wait the appropriate delay time, *f*) check if the registration was successful.

In the table 6.3 we present results of conducted time tests are presented.

Differences in times between Vodafone and O2 networks could happen due to external distortions or longer distance from a BTS station.

All tests proved that the cumulated allowed delay's time is equal to 7,5 seconds. Taking into consideration the fact that the average substitution time is equal to 4,7 seconds it is enough time to conduct the attack. It is also possible to decrease this time even more (below 4,7 seconds). Propositions of various actions which may led to achieving this goal are described in the following section.

6.1.4 Improving speed of the attack

There are several ways to improve speed of our implementation of the man-in-the-middle attack. We may start from writing own implementation of the SAP server which will use a standalone smart card reader. In such a way the attacker won't use the Bluetooth connection. Access time of reading the data with usage of the standalone

smart card's reader will be comparable to the time necessary to obtain the adequate data from regular phone. It will be much faster than the Bluetooth connection.

Second option is fake phone implementation done in C++ instead of the Python programming language. It will increase the speed but will also need greater development efforts and much more time. It may require complicated debugging phase and cause possible problems with drivers.

Another proposition can be rewriting the Telit CMUX drivers using the information from [Com09a] in order to make them work in the Linux environment. Attacker will get rid of the necessity to use Ethernet to exchange the data between the fake phone and the fake BTS. After doing that, the fake phone and the fake BTS may be realized in the form of communicating processes running on the same machine. In case of rewriting the fake phone application it may even be merged with the fake BTS and become a part of the OpenBTS application.

We may also try to do substitution of the Telit GT864-PY mobile station with, for example, osmocom [3] module should give the attacker better control over the connection speed. It will also allow to modify easily the IMEIcode, which may improve the reliability of the attack. In case of too many connections from different IMSIs from device with the same IMEI it may work as the warning sign for MNO and lead to the discovery of the attack.

6.2 A5/2 support

Information in this section is based on: [Wel10], [Sec11], [rGPPG], [BBK07]. The A5/2 was designed as a weaker version of the A5/1 cipher in order to sell it to countries from the outside of western Europe (while the A5/1 had strong export restrictions). The cipher's internal architecture, created in the late 1980s, was kept secret for more than 10 years. It was provided only on a need-to-know basis, under an non-disclosure agreement, to the GSM manufacturers. The idea behind that was to provide *security by obscurity*. The internal design of both the A5/1 and the A5/2 was reverse engineered from an actual GSM phone by Briceno [BGW99] in 1999. It was verified against known test-vectors, and it is available on [BGW99]. Organizations responsible for the GSM (ETSI/3GPP/GSMA) decided to organize special research groups, which goal was to evaluate a possible risk of using the A5/2 cipher, in 2003, nearly 4 years later. During this time in several scientific publications it was proven that not only there exist successful attacks against A5/2 ([GWG99], [PFS00]) but also that an attacker may decrypt a recorded A5/1 transmission, because the A5/2 uses the same keys as the A5/1. Only solution was to deprecate and later remove the A5/2 support from all the equipment. Starting from 2004 3GPP and GSM delegated special groups to undertake appropriate actions, as described in [oGTSW04]. Those groups managed, in the following years (up to 2007), to convince their respective bodies (3GPP, GSMA), and their members (e.g. operators, equipment manufacturers) to officially decide about removal of the support of A5/2 from GSM networks.

3. OsmocomBB `http://bb.osmocom.org/trac/` [Online, last check 21.03.2011]

The reaction time was very poor. It took eight years to officially fix the problem. Biggest opposition against removing the A5/2 cipher support was in North America. It is quite interesting since United States operators always had access to the A5/1. In July 2007, the 3GPP has approved a request to prohibit the implementation of A5/2 in new mobile phones. These events showed almost no interest of GSM equipment's manufacturers and mobile operators in removing existing security holes. There is also no plan of upgrading or patching the entire system - no proactive security response plans. Detailed time-line with documents extending the topic is available at [Sec11]. In theory all phones produced after 2007 do not support the A5/2 cipher. In the laboratory we conducted tests on various phones to check in practice if A5/2 is still supported by manufacturers. Results are presented in Table 6.4.

Test results proved that all tested phones which were designed for the German market and produced after 2007 have switched-off support for the A5/2. MyPhone which was designed in China still supports the A5/2. Therefore we may conclude that the man-in-the-middle attack can affect all phones in countries in which this type of ciphering is still in use. Test results in Poland may be different because producers often admit that they are selling here products which are designed for developing markets. In other words the fact that one of the phone does not support the A5/2 in Germany does not mean that the same model in version for polish market does not support this cipher either.

6.3 Financial costs

In this section present an approximate price of the man in the middle attack implemented with designed by us devices and scripts. The approximate equipment cost of the attack can be calculated several ways. Total cost, which assume that attacker has to buy also laptops and Windows license, is presented in the Table 6.5.

Item	Prices in Euro	Source
Modified USRP	~2500	http://kestrelsignalprocessing.mybigcommerce.com/products/OpenBTS-Development-Kit.html [Online, last check 21.03.2011]
Telit GT864-PY	~150	http://www.advoco.ie/telit-gt864-py-terminal.html [Online, last check 21.03.2011]
Samsung Corby	~100	http://www.computeruniverse.net/products/90377656/samsung-corby-3g-s3370-silber.asp?agent=619 [Online, last check 21.03.2011]
Windows 7 Home	~80	http://www.preis.de/produkte/Microsoft-MS-Windows-\7-Home-Premium-64-Bit/539283.html [Online, last check 21.03.2011]
2 MEDION AKOYA S3212 laptops	800	http://www.medion.com/de/electronics/prod/MEDION%C2%AE+++AKOYA%C2%AE+S3212+(MD+98150)+/30010608B [Online, last check 21.03.2011]
Total	3630	

Table 6.5: The man in the middle attack's equipment costs summary

In case if an attacker has already access to Windows license and computers capable of running the fake phone and fake BTS application the total cost may reduce to the price of the USRP, Samsung Corby and Telit module and will be equal to 2750 Euros. If he has additionally skilled in electronics instead of the modified USRP he may buy its regular version and do the modifiction on his own. Calculation of the approximate price of this scenario is presented in Table 6.6.

Item	Prices in Euro	Source
USRP	~500	http://www.ettus.com/order [Online, last check 21.03.2011]
RFX1800 - 1.5 to 2.05 GHz Transceiverboard for USRP	~200	http://www.ettus.com/order [Online, last check 21.03.2011]
52MHz clock board - FA-Synthesizer 'FA-SY 1', 10 - 160 MHz, CMOS	40	http://www.box73.de/catalog/product_info.php?products_id=1869 [Online, last check 21.03.2011]
Telit GT864-PY	~150	http://www.advoco.ie/telit-gt864-py-terminal.html [Online, last check 21.03.2011]
Samsung Corby	~100	http://www.computeruniverse.net/products/90377656/samsung-corby-3g-s3370-silber.asp?agent=619 [Online, last check 21.03.2011]
Total	990	

Table 6.6: The man in the middle attack's minimal equipment costs approximation

Additionally one may take into the consideration also the price of programmer's work. The implementation, done basing on the full technical report, should take approximately 10 working days. Each working day is equal to 8 hours. We may set the worker hourly wage to the average 15 euro for a qualified, educated engineer. That gives us together an additional 1200 euro. The full attack's price is then equal to more or less 5 000 euros.

All prices propositions sets the cost of the attack within the financial abilities of most of the qualified telecommunication engineers. The necessary equipment may realistically be build by a skilled amateur even as a hobby. That increases probability of the attack and makes it more dangerous.

6.4 Possible attack scenarios

Man-in-the-middle attack on the Um interface of the GSM network may be done in form of several attack scenarios. In this chapter we will present most important of them: *a*) call wire-tapping, *b*) call hijacking, *c*) altering of data messages (SMS), *d*) call theft – dynamic SIM card cloning.

In each of the scenarios attackers forces a victim's mobile station to connect to a mobile network operator through his own equipment. Later he may eavesdrop, modify, block, or create victim's data. Wide spectrum of the possibilities of malicious ways in which the attack may be used against victim proves how dangerous it can be. Scenarios proposed below are only the basis for later modifications and tuning. In this chapter

we present the evaluation of the most probable attacker's actions and risks arising from them.

The biggest possible pitfall of all these scenarios would be situation in which the victim somehow escapes from the fake BTS network's range, for example by moving to another place, and the real GSM network will receive two registrations from the same identity. Excluding this situation, a properly conducted attack is almost impossible to detect for a GSM network and the victim. Note that most operators do not, in fact, check for duplicate IMSI registrations making the attack practically undetectable. Descriptions of scenarios are based on [BBK07].

6.4.1 Call theft

This attack scenario was implemented in the test environment in order to present a live, real time proof of the attack. It is presented on the Figure 6.2

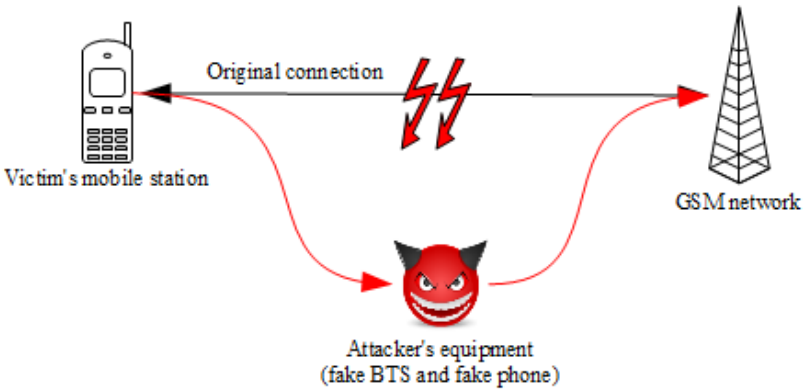

Figure 6.2: Man-in-the-middle attack general scenario

In this scenario the attacker uses victim's mobile station only at the beginning of the communication with the GSM network. He is doing it in order to authenticate his own mobile station (fake phone). Later, using the stolen credentials, he may undertake all possible actions allowed for the victim in the GSM network. By impersonating the victim he may send and receive data, make his own phone calls, send and receive SMS messages. Attacker may also try to extend the time of the victim's registration session. The mobile station of the victim is used in this scenario only as the oracle for obtaining authentication data (SRES and Kc). Detection of this attack is very hard. It is so because from the GSM network perspective everything looks like regular procedures of connection and authentication. Also the victim in case of the data forwarding does not have any chance to realize that the attack is ongoing.

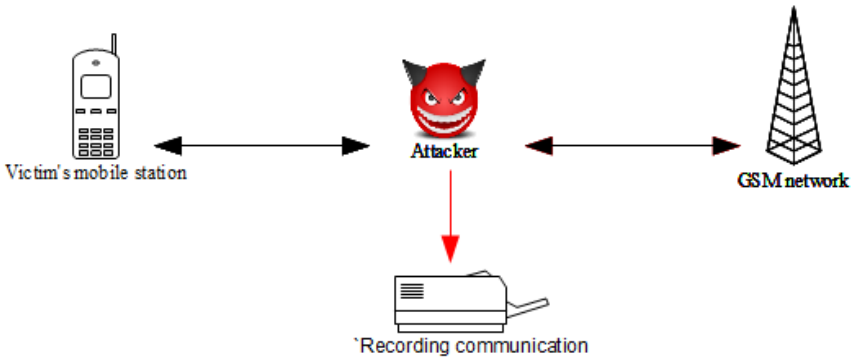

Figure 6.3: Call Wire Tapping scenario

6.4.2 Call wire

This is the most intuitive scenario. It is goal is to capture and record for further usage all the data sent from the victim's phone to the GSM network. The attacker does not interfere the transmission, he only forwards information. Scenario is presented on the Figure 6.3. During this scenario the attacker may also decrypt communication and eavesdrop conversations in real-time. He may also eavesdrop all kinds of exchanged data for example SMS messages. Another possible usage of this scenario is to record all the victim's traffic and later decrypt it. It should be easier to implement and should also consume less resources. Such an approach would allow to eavesdrop several conversations during the same time on one machine.

6.4.3 Call hijacking

In this scenario an attacker can cutoff the victim's ongoing conversation, and later impersonate the victim to the conversation's responder. That may happen at a very early stage of the conversation - even before the victim's mobile phone will ring. Detecting this malicious actions by an operator is hardly possible. The only clue can be the increased electro-magnetic interference.

6.4.4 Altering of data messages (SMS)

This scenario starts from hijacking the transmission between a victim's mobile station and the GSM network by the attacker. Later he may take full advantage of the controlled connection's channel and eavesdrop the content of victim's SMS messages, modify them, block or even create his own ones. In such a way he compromises the integrity of the GSM traffic. As in the case of previous scenarios this scenario, if conducted properly, is very hard to detect. Its practical implementation may allow the attacker for example to send messages to paid services owned by him on expense of the victim. The general schematics of this attack scenario is presented on the Figure 6.4.

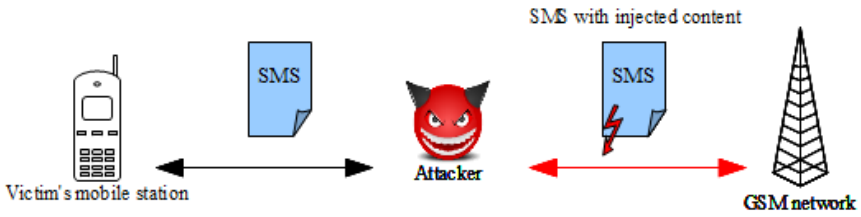

Figure 6.4: Altering by an attacker of the SMS sent from a victim

Model	On the market since	Product website	A5/2 support ?
HTC Touch Viva	2009	`http://www.htc.com/www/product/touchviva/overview.html`	NO
BlackBerry Curve 8300	2007	`http://worldwide.blackberry.com/blackberrycurve/`	NO
Motorola RAZR V3xx	2006	`http://www.gsmarena.com/motorola_razr_v3xx-1648.php`	YES
LG GM360	2010	`http://www.gsmarena.com/lg_gm360_viewty_snap-3359.php`	NO
Nokia 6300	announced in 2006, released in 2007	`http://www.gsmarena.com/nokia_6300-1800.php`	YES
Samsung Corby	2009	`http://www.gsmarena.com/samsung_s3650_corby-2908.php`	NO
MyPhone 6691	2009	`http://www.mgsm.pl/pl/katalog/myphone/6691/myPhone_6691.html`	YES

Table 6.4: Results of the test checking support of A5/2 by various mobile phones

7 Conclusion

In the thesis we proved, in the form of a working implementation, that it is possible to conduct a man-in-the attack on the Um interface of a GSM network. The thesis result has been confirmed by a live presentation. Moreover, all equipment used in the project was amateur or GSM general purpose. There was no need to use expensive modifications or expensive equipment in order to achieve the goal. We proved that an attacker is able to bypass all GSM limitations and security restrictions and conduct dynamic SIM card cloning. Our live demonstration finished with sending an SMS message on the expense of a victim, but there exist a wide variety of possible attack's malicious applications [1]. It clearly showed that the A5/2 cipher is no longer secure solution and it is relatively easy to use its weaknesses against stronger A5/1.

In consequence, mobile telephony users should be more careful than ever. The skilled attacker using open source solutions can cause a serious damage and compromise the integrity of a communication channel. He may not only attack regular mobile stations but also all the equipment communicating with the usage of the Um interface; for example, ticket selling machines.

There are several ways to reduce the attack's impact. The most intuitive approach would be to stop using the A5/2 cipher in any mobile equipment, if it is only possible. This approach is increasingly selected, especially by manufacturers from developed countries such as EU or USA. Unfortunately not all countries and GSM network operators may use A5/1 encryption and they wont be able to adjust. [2]. The second intuitive solution could be a standards change to substitute A5/2 cipher with a newer or better implementation. It is the most radical approach, since it would force the producers to change technological lines and would prevent owners of older mobile stations from connecting to the network. In this scenario one may also add a mechanism which would allow the mobile station to check the authenticity of the GSM network to which it is connected. A partial temporary solution would be to decrease mobile station allowed registration time [3] wherever it is possible. In such a way an attacker simply may not have enough time to conduct the attack. Strict control of the number of registrations of each IMSI should also become widespread.

This project can be further developed in several directions, making the attack more dangerous and harder to discover. We may implement all of the attack scenarios [4] and test them on wide variety of equipment, not only mobile phones, for example ticket selling machines or parking meters. We may improve our software to decrease even further the speed of the attack – detailed information in Section 6.1.4 or create more

1. More information in Section 6.4
2. Detailed information about the support of A5/2 cipher, including tests, is available in Section 6.2
3. More information in Section 6.1.4
4. Detailed description in Section 6.4

intuitive graphical user interface. Finally we may experiment with the victim acquisition described in detail in Section 2.4.2, for example by testing our USRP in different environments with different antennas.

Bibliography

[3GP97a] 3GPP/ETSI. Gsm 02.07 mobile station (ms) features. http://www.3gpp.org/ftp/specs/html-info/0207.htm, 1997. 12

[3GP97b] 3GPP/ETSI. Gsm 02.09 security aspects. http://www.3gpp.org/ftp/specs/html-info/0209.htm, 1997. 8

[3GP98a] 3GPP/ETSI. 3gpp ts 03.20 security-related network functions. http://www.3gpp.org/ftp/specs/html-info/0320.htm, 1998. 9, 50

[3GP98b] 3GPP/ETSI. 3gpp ts 04.08 mobile radio interface layer 3 specification. http://www.3gpp.org/ftp/specs/html-info/0408.htm, 1998. 9, 47, 50, 51

[3GP98c] 3GPP/ETSI. 3gpp ts 07.07 at command set for gsm mobile equipment (me). http://www.3gpp.org/ftp/specs/html-info/0707.htm, 1998. 31

[3GP00] 3GPP/ETSI. 3gpp ts 07.05 use of data terminal equipment - data circuit terminating equipment (dte-dce) interface for short message services (sms) and cell broadcast services (cbs). http://www.3gpp.org/ftp/specs/html-info/0705.htm, 2000. 31

[Apv10] Axelle Apvrille. *OpenBTS for dummies - v0.2*. 2010. 39, 41

[BBK07] Elad Barkan, Eli Biham, and Nathan Keller. Instant ciphertext-only cryptanalysis of gsm encrypted communication. Computer Science Department Technion – Israel Institute of Technology Haifa, 2007. xi, xiii, 8, 16, 17, 18, 19, 25, 26, 37, 51, 62, 66

[BGW99] Marc Briceno, Ian Goldberg, and David Wagner. A pedagogical implementation of the gsm a5/1 and a5/2 voiceprivacy encryption algorithms. http://cryptome.org/gsm-a512.htm, 1999. 16, 25, 62

[Bil11] Kai Billen. Imsi-catcher - wanzen fur handys. http://hp.kairaven.de/miniwahr/imsi.html, 2011. xi, 26

[Com04] Telit Communications. Telit modules software user guide. http://www.telit.com/module/infopool/download.php?id=522, 2004. 48, 50, 52, 59

[Com08] Telit Communications. Gt864-quad / py terminal product description. http://www.telit.com/module/infopool/download.php?id=555, 2008. xi, 31, 48

[Com09a] Telit Communications. Cmux user guide. http://www.telit.com/module/infopool/download.php?id=616, 2009. xi, 31, 32, 34, 48, 62

[Com09b] Telit Communications. Sim access profile user guide. http://www.telit.com/module/infopool/download.php?id=522, 2009. 32, 48, 49, 50

[ETS] European Telecommunications Standards Institute ETSI. `http://www.etsi.org/`. 5

[EVBH09] Jorg Eberspocher, Hand-Jorg Vogel, Christian Bettstetter, and Christian Hartmann. Gsm architecture, protocols and services. Wiley, 2009. xi, 8, 22

[GSM09] GSMA. Market data summary. `http://www.gsmworld.com/newsroom/market-data/market_data_summary.htm`, 2009. 5

[GSM11] Gsm for dummies. `http://gsmfordummies.com/`, 2011. 8

[GWG99] Ian Goldberg, David Wagner, and Lucky Green. The (real-time) cryptanalysis of a5/2. Rump Session of Crypto'99, 1999. 16, 25, 62

[LLC09] Ettus Research LLC. Brochure for the entire usrp product family. `http://www.ettus.com/downloads/ettus_broch_trifold_v7b.pdf`, 2009. 39

[oGTSW04] Secretary of 3GPP TSG-SA WG3. Draft report of sa3 meeting 36. `http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_36_Shenzhen/Report/Draft_Rep_v004_SA3_36.pdf`, 2004. 62

[PFS00] Slobodan Petrovic and Amparo Fuster-Sabater. Cryptanalysis of the a5/2 algorithm. IACR ePrint Report 2000/052, 2000. 25, 26, 62

[rGPPG] 3rd Generation Partnership Project (3GPP). `http://www.3gpp.org/`. 62

[rGPPGT99] 3rd Generation Partnership Project. 3GPP TS 11.12. Specification of the subscriber identity module -mobile equipment (sim-me) interface. `http://www.3gpp.org/ftp/specs/html-info/1112.htm`, 1999. 49, 50

[SB11] Harvind Samra and David A. Burgess. Um interface. `http://en.wikipedia.org/w/index.php?title=Um_interface&oldid=417988078`, 2011. 8

[Sec11] Osmocom Security. Withdrawal of a5/2 algorithim support. `http://security.osmocom.org/trac/wiki/A52_Withdrawal`, 2011. 62, 63

[SIG08] Bluetooth SIG. Sim access profile. `https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=158740`, 2008. 31, 32

[Weh09] Dennis Wehrle. Master-arbeit zum thema: Open source imsi-catcher. Albert-Ludwig-Universitat Freiburg, 2009. xi, 14, 21, 40, 42

[Wel10] Harald Welte. A brief history on the withdrawal of the a5/2 ciphering algorithm. `http://laforge.gnumonks.org/weblog/2010/11/12/`, 2010. 62