Call My Number Instead: Man in the Middle Attacks in GSM Despite Mutual Authentication

Anonymous Author(s)

ABSTRACT

The Global System for Mobile Communications (GSM) provides confidentiality protection over the air interface. Publicly described attacks to break confidentiality of circuit switched voice calls in GSM are cryptographic attacks on weak ciphers, or fake base stations performing a Man-in-the-Middle (MitM) attack to impersonate a network. As a result, countermeasures such as stronger cryptographic algorithms with longer keys and mutual authentication using UMTS AKA in GSM have been standardized. This paper presents an attack on the air interface that works despite mutual authentication and stronger cryptography. The core of this attack lies in the lack of integrity protection. A man in the middle attacker during mobile originated call setup can change the encrypted call setup message to redirect the call to himself and then forward the call to the intended destination. This way, the attacker will receive the cleartext communication without having to continue to be a man in the middle on the air interface. This attack was demonstrated with a virtual physical layer, which was developed for the open source projects Osmocom and OsmocomBB.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**;

KEYWORDS

GSM, confidentiality

ACM Reference Format:
Anonymous Author(s). 2018. Call My Number Instead: Man in the Middle Attacks in GSM Despite Mutual Authentication. In *Proceedings of ACM Conference on Computer and Communications Security (CCS2018).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Since introduction of GSM as the second generation (2G) of cellular communication standard in 1992, cellular communication system were constantly improved. New, faster and more secure standards were specified and deployed. Because the existing GSM network has broad coverage, because its infrastructure is already deployed, and because it is inexpensive to operate, GSM is still in use despite the roll out of 3G and 4G networks [25]. Some countries have already switched off their 2G networks, however in other countries GSM is still operational and may remain in operation for a number of years to come[9], [19].

When it comes to security, GSM is operated today without major changes to the state of the art from the time it was specified 30 years ago. Despite some attempts at increasing security such as introduction of mutual authentication between network and mobile phone (in GSM language: mobile station, MS) and better encryption such as A5/4 into the standards, the deployments are still vulnerable to a number of attacks.

Reasons for this are slow adoption of new technology when security isn't the customers primary goal and the operator's desire to support customers with old handsets. More fundamentally, GSM was designed at the end of the 1980s with an expected lifetime of 20 years. The attacker model at that time was one of passive attacks. Nowadays, active attacks have become affordable and can be realized at a low cost. In the standards, this had been foreseen already during the development of the 3rd generation of cellular communication systems Universal Mobile Telecommunications System (UMTS) and taken into account accordingly. Some of the improvements designed into UMTS were then ported back into GSM.

For example, introduction of UMTS-Authentication and Key Agreement (AKA) into GSM allows a mobile station (MS) to authenticate the network, and by association, the base station (Base Transceiver Station (BTS)). One of the new features of UMTS that wasn't ported back into GSM was integrity protection of the control channels. This is exploited by our attack on mobile originating calls: if the attacker has knowledge of the called party's phone number, the attacker can modify the ciphertext of the call setup message to redirect the call by changing the target phone number to a phone number under control of the attacker. The attacker can then listen in to the communication at that phone number and forward the call to the originally intended called party. The assumption that an attacker knows the called party's number may sound drastic, but there are many situation in which this is not as far fetched. Examples could be a client calling her lawyer, an undercover agent calling his handler, a CEO calling the secretary etc.

The assumption that the attacker knows the called party's number can be relaxed if the attacker controls a large block of numbers with a constant prefix. Then all the attacker would have to guess is the prefix of the called number, e.g. the country code. So this attack could be mounted by someone who has control over or close contacts with a network operator, e.g. in a foreign country.

The contribution of this paper is twofold: firstly, it validates the proposed man in the middle attack. Secondly, it describes a virtual air interface for osmoCOM[1] that can be used for research purposes as well.

The paper is structured as follows: in the next Section, an overview of the GSM system is given. In the following two Sections, the parts

CCS2018, October 2018, Toronto, Canada

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

[1] https://osmocom.org

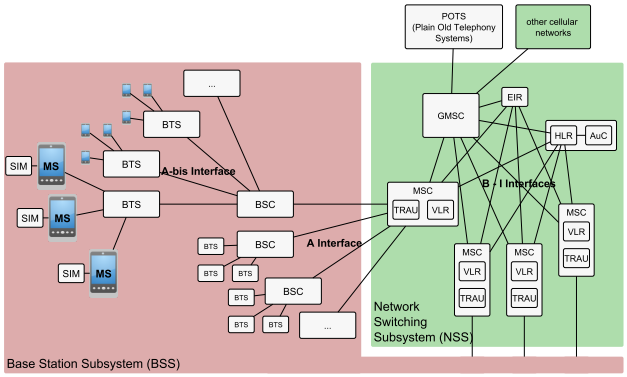

Figure 1: GSM network architecture, cf. Schnabel [28]

of the GSM system relevant for the attack presented in this paper are further detailed: the call setup in Section III and processing for the air interface in Section IV. Section V goes into detail of the attack itself and potential countermeasures. In Section VI the implementation and verification of the attack within the osmoCOM project is described. Section VII gives an overview of related work. The final Section presents a conclusion and an outlook on future work.

2 GSM OVERVIEW

2.1 GSM Architecture

The GSM-network has a hierarchical architecture, as depicted in Figure 1. Components for packet switching (General Packet Radio Service (GPRS)) and interfaces to 3G, 4G and other access technologies are not shown. The Base Station Subsystem (BSS) manages and operates the air interface for transmitting voice and data from and to the MS. The Network Switching Subsystem (NSS) relays voice and data within an operator or to gateways to other cellular or fixed line operators.

2.1.1 MS - Mobile Station. The MS denotes the mobile device that accesses the network. The Subscriber Identity Module (SIM) card is part of the MS. The SIM card contains the International Mobile Subscriber Identity (IMSI) and secret key of the subscriber. The operator assigned IMSI uniquely identifies the subscriber, and is used to allow an operator to bill services used by the subscriber to that subscriber. The secret key Ki never leaves the SIM-card and is only used for authentication and key generation by the algorithms A3 and A8 which are implemented on the SIM card itself.

2.1.2 BTS - Base Transceiver Station. The BTS communicates with the MS's over the air interfaces. Uplink and downlink frequencies are separated by a fixed offset (Frequency Division Duplex, FDD) and uniquely identified by an Absolute Radio Frequency Channel Number (ARFCN). Neighboring BTS are assigned different frequencies to avoid interference. Multiple MS's may be connected to the same BTS simultaneously Multiplexing between MS is done by Time Division Multiple Access (TDMA), in which one of eight time slots in each frame is assigned to a MS.

2.1.3 BSC - Base Station Controller. The BSC manages multiple BTS's via A-bis Interfaces. It relays data and speech between the NSS and the appropriate BTS.

2.1.4 MSC - Mobile Switching Center. Multiple BSCs are connected to the core network and other networks via an MSC. The MSC is also part of the Signaling System 7 (SS7) network.

2.1.5 HLR - Home Location Register. The HLR contains a data base with information of all subscribers of a provider, such as contract details, network access permissions, prepaid balance, Mobile Subscriber Routing Number (MSRN), Mobile Subscriber ISDN Number (MSISDN) and current VLR. It also contains the Authentication Center (AuC) that has all secret key in GSM (Ki) for the subscribers and implements A3, the algorithm used to calculate the expected response (RES) and A8, the algorithm used to calculate the shared key (Kc) used for all ciphering between network and MS. The calculation of repsonse and ciphering key takes as input a challenge, so the keys are not reused. Thus Ki never has to leave the AuC, only derived key Kc and authentication information RES are shared.

2.1.6 VLR - Visitor Location Register. The VLR is usually part of the MSC and keeps a copy of the mobility information (such as the Location Area Index (LAI)) from the HLR for the MS's connected to this MSC. For these MS, it also contains the mapping of Temporary IMSI (TMSI) to the IMSI and MSRN. The VLR requires the address of the HLR to forward authentication requests via the SS7 network.

2.2 Um Interface

The air interface between MS and BTS is called Um-interface or simply Um. On physical layer, a combination of Frequency Division Multiple Access (FDMA) and TDMA is used to create multiple channels. Each carrier frequency is split into 8 physical TDMA channels. A physical channel can be split into several logical channels, again by using TDMA, i.e. by assigning a periodic sequence of time slots, a so called multi frame. Different logical channels are used for different functions. For example, Traffic Channel (TCH) is used for voice, Broadcast Control Channel (BCCH) is used for broadcast control information and Common Control Channel (CCCH) is used for control data for individual subscribers.

Within one timeslot of a physical channel exactly one burst is transmitted. Figure 2 shows the format of a regular burst. Regular bursts are used in GSM both for transmission of voice data (i.e. TCHs) as well as signaling (e.g. Standalone Dedicated Control Channel (SDCCH), Slow Associated Control Channel (SACCH) or Fast Associated Control Channel (FACCH)). Each burst carries two blocks of 57bit. Training sequence, front and rear tail are relevant for demodulation. The stealing flags can be used to indicate that blocks of the TCH burst are used for control information [31, subclause 5.2.3].

3 CALL SETUP

The attack works by interfering with the mobile originating call setup procedure. Therefore, this section introduces the normal setup procedure as it presents itself on the air interface and indicates the weak point that this attack is exploiting.

Figure 3 gives the complete message sequence over the air (Um interface). The message sequence chart here includes all layers over

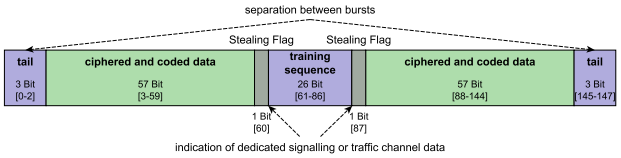

Figure 2: Packet format of a regular burst, according to 3GPP TS05.02 [31, subclause 5.2.3]

Um, giving the channel the messages are being sent over, as well as which protocols are involved. Thus, this chart is aligned with what sniffing on the air interface with wireshark would yield, e.g. using a Nokia 3310 and GAMMU[2].

For a mobile originating call, the MS first requests a radio resource for a dedicated signaling channel. This happens with a short message on the Random Access Channel (RACH). The base station uses this access burst to determine at what time the MS has to send the following bursts in order for the burst to arrive at the BTS within its allocated time slot. The network assigns the requested channel, and the MS sets it to be reliable with the SABM procedure. After the SABM command has been executed, every message on layer two, i.e. Link Access Procedure for the Dm-Channel (LAPDm), will be acknowledged. On layer two, for every frame sent, a timer is set, and the frame is resent if the acknowledgment is not received in time. In Figure 3, acknowledgments are indicated by the Radio Resource (RR) messages on LAPDm.

After that, the network can request the MS identity. Even though this is optional, the network normally requests the identity, as otherwise the network has no way of attributing and thus billing the call. The MS can respond with either its permanent identity IMSI, or its temporary identity TMSI.

The third step, i.e. authentication, is optional as well. In authentication, the network sends a random number to the MS. The MS, or more precisely, the SIM card, calculates from this function a response RES and a key Kc. RES is returned to the network which can verify that this is the expected value. The network may skip the authentication step and reuse Kc that was calculated from a previous authentication run.

The fourth step has the network starting the encryption with a cipher mode command, which includes the algorithm to be used for encryption. This step is optional, because some jurisdictions did not permit encryption. Most networks today actually set up encryption.

In the fifth step, the MS finally informs the network that it would like to make a call, using the SETUP message of the call control protocol. It is this message that contains the destination phone number, which is the message we are trying to modify. The network returns an acknowledgment on LAPDm and then, after verifying that the parameters of the SETUP message, such as the phone number, are valid, it sends a call processing message, which again is acknowledged on a lower layer. Otherwise it sends an error message that causes the mobile to clear the call, i.e. abandon the call attempt. The call SETUP message is supervised by a timer with a value of

[2]cf. https://wammu.eu/gammu/

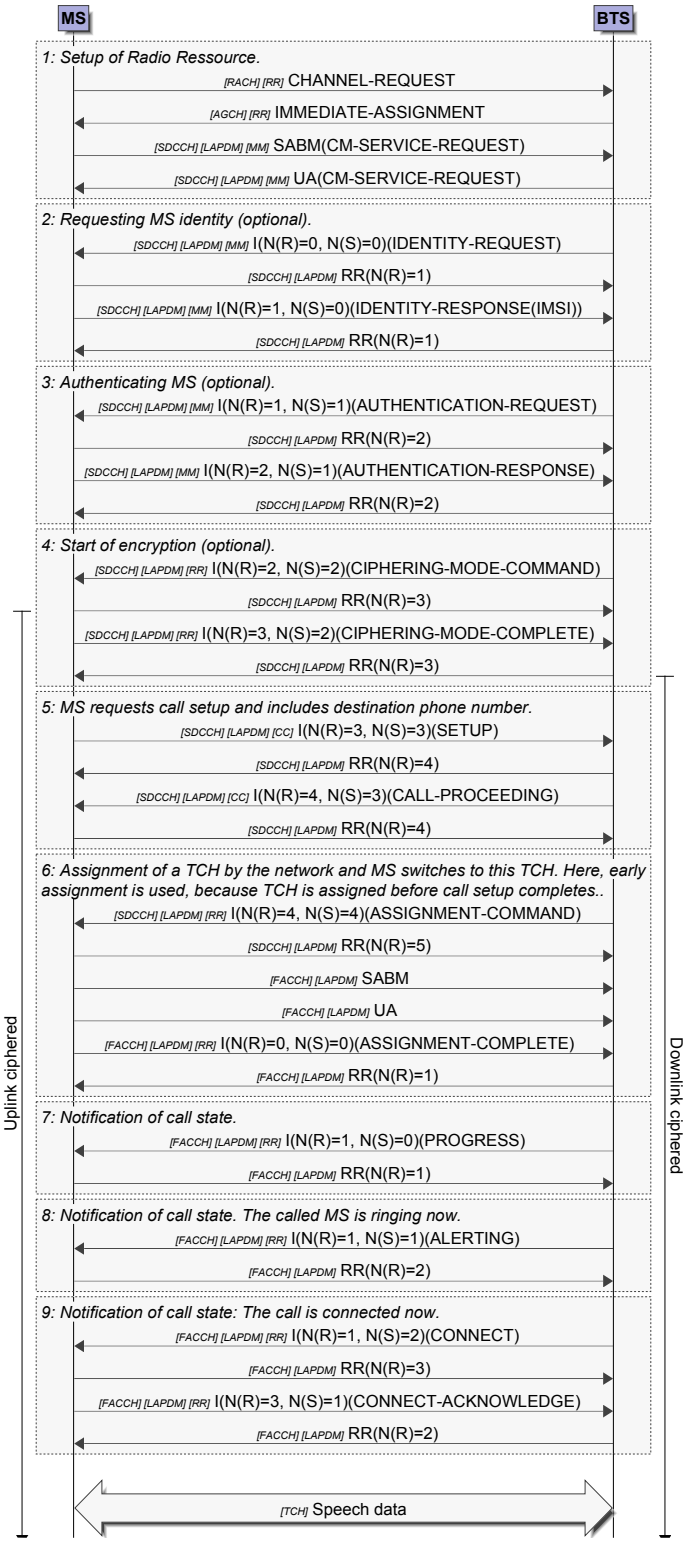

Figure 3: Mobile originating call setup. Channel and protocols are indicated in square brackets []

30 seconds, and in case the call proceeding message is not received until the timer expires, the mobile clears its state and abandons the call attempt. This means the only retransmit mechanism available is on layer 2.

Step six assigns a traffic channel and sets up reliable data transfer for the associated FACCH. In step seven and eight, the network keeps the mobile informed of the progress the network is making in getting the call connected. Step nine then informs the mobile that the call is connected, and from then the voice call data is exchanged over the TCH.

4 SIGNAL PROCESSING AND CHANNEL CODING

In GSM the physical layer is responsible for signal processing, coding, encryption and modulation. Speech is compressed with a lossy compression algorithm to lower the bandwidth required. No compression is standardized for other types of data.

Different channels have different requirements and are therefore treated differently. Figure 4 shows the different coding schemes applied. For signaling data on a channel that the call setup message is on, such as SDCCH and FACCH, first a fire code is added to aid error detection. The result is coded with a half rate convolutional code to provide forward error correction. Finally, the resulting bits are mapped onto bursts such that the impact of noise distorting neighboring bits in the burst is reduced. The bursts are then encrypted and modulated onto a carrier frequency.

4.1 Block code

A block code adds redundancy to a block of data in order to detect or correct errors. GSM defines two mechanisms, Cyclic Redundancy Check (CRC) for data on traffic channels and fire code for signaling channels.

The latter is of interest for our attack. A fire code is a linear block code that will return a number of redundancy bits for a message block. Calculation of the redundancy bits are done like with a CRC. The message is interpreted as a polynomial with each bit representing one coefficient. This is divided by the generator polynomial of the fire code. The redundancy bits are produced by the remainder of the polynomial long division. Simply appending the remainder bits to the message will yield a remainder of zero when performing the polynomial long division. XORing the remainder with a bit pattern before appending will lead to that bit pattern being the remainder of the long division when verifying. In GSM, the generator polynomial (cf. Equation 1) and the desired remainder $\mathbf{r}$ of the long division ($\mathbf{r}$ = `0xfffffffffff`) are defined in 3GPP TS05.03[32, Subclause 4.1.2].

$$\begin{aligned} g(x) &= (x^{23}+1)\cdot(x^{17}+x^3+1) \\ &= x^{40}+x^{26}+x^{23}+x^{17}+x^3+1 \end{aligned} \tag{1}$$

Input for block coding is a LAPDm frame $\mathbf{d} = \mathbf{d}(1)\ldots\mathbf{d}(184)$ with a fixed size of 184 bit. By polynomial long division with the Fire code generating polynomial g, 40 redundancy bits, $\mathbf{p} = \mathbf{p}(1)\ldots\mathbf{p}(40)$ are produced. These are then XORed with the expected remainder $\mathbf{r}$ to give $\mathbf{p}_r = \mathbf{p}_r(1)\ldots\mathbf{p}_r(40)$ which is appended to the input. The result is then padded with four bit of value zero. The output of block coding, $\mathbf{u}_r$, is given in Equation 2. With the generator polynomial, it would be possible to detect and correct burst errors of up to 11-bit [10]. GSM, however, only makes use of error detection and relies on retransmission of the LAPDm frame for error correction.

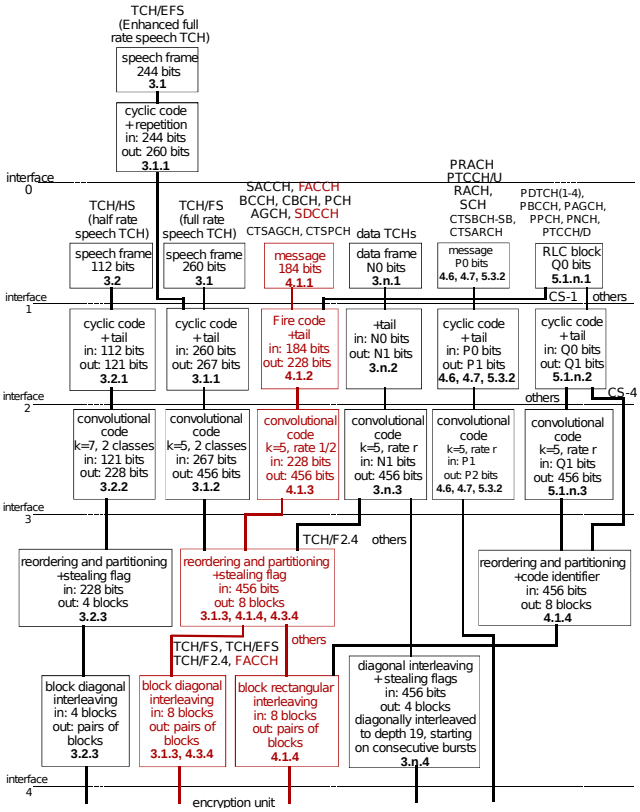

Figure 4: Channel coding in GSM, taken from 3GPP TS05.03[32, Figure 1a]. It includes references to the subclause in TS05.03 in which more detail is given.

$$\begin{aligned} \mathbf{u}_r(k) &= \mathbf{d}(k) && \text{for } k = 0, 1, \ldots, 183 \\ \mathbf{u}_r(k) &= \mathbf{p}_r(k-184) && \text{for } k = 184, 185, \ldots, 223 \\ \mathbf{u}_r(k) &= 0 && \text{for } k = 224, 225, 226, 227 \end{aligned} \tag{2}$$

The Fire code doesn't include any secret in its calculation and as thus doesn't provide integrity protection against a man in the middle who can recalculate the redundancy bits.

4.2 Convolutional code

The convolutional code adds redundancy for forward error correction. Like block codes, the convolutional code can be described in terms of a generator polynomial that is convoluted with the data.

The generator polynomials of the $\frac{1}{2}$ rate convolutional code used in GSM is given in the standard in TS05.03[32, Subclause 4.1.3] as g_0, g_1 (cf. Equations 3)

$$g_0 = x^4 + x^3 + 1$$
$$g_1 = x^4 + x^3 + x + 1 \quad (3)$$

These equations mean that for each input bit x two bits are output, g_0 and g_1 as defined by XOR of x with some of its predecessor bits. When no predecessor bits exist, they are assumed to be 0.

Input to convolutional coding is the output of the blockcode. The result of convolutional coding $\mathbf{c}(k)$ is expressed in Equation 4.

$$\begin{aligned} \mathbf{c}(2k) &= \mathbf{u}_r(k-4) \oplus \mathbf{u}_r(k-3) \oplus \mathbf{u}_r(k) \\ \mathbf{c}(2k+1) &= \mathbf{u}_r(k-4) \oplus \mathbf{u}_r(k-3) \oplus \mathbf{u}_r(k-1) \oplus \mathbf{u}_r(k) \\ &\text{with } k = 0, 1, \ldots, 227 \\ &\text{and } \mathbf{u}_r(j) = 0 \quad \forall\, j < 0 \end{aligned} \quad (4)$$

Again, no secret is involved in calculating the convolutional code thus it does not provide integrity protection against MitM attacks. Furthermore, the convolutional code is linear, i.e. Equation 5 holds.

$$\boldsymbol{c}(\boldsymbol{x} \oplus \boldsymbol{y}) = \boldsymbol{c}(\boldsymbol{x}) \oplus \boldsymbol{c}(\boldsymbol{y}) \quad (5)$$

4.3 Interleaving

To protect against burst errors, data is reshuffled by the interleaver. Bits affected by burst errors will be non-consecutive after de-interleaving, thus increasing the probability of correcting the errors. Each data blocks N_n of 456 bit coming out of the convolutional coder is split into 8 blocks of 57 bit each. every burst B_b is made up of two such blocks.

In a traffic channel, these 8 blocks are interleaved over 8 bursts such that every byte has one bit in each burst. For dedicated control channels, the interleaving is such that 8 blocks are interleaved over 4 bursts [32, subclause 4.1.4].

$$\begin{aligned} &i(B_b, j) = c(N_n, k) \\ &k = 0, 1, \ldots, 455 \\ &n = 0, 1, \ldots \\ &b = 4 \cdot n + (k \bmod 4) \\ &j = 2 \cdot ((49 \cdot k) \bmod 57) + ((k \bmod 8) \operatorname{div} 4) \end{aligned} \quad (6)$$

The interleaving does not depend on any of the data being interleaved, nor is there a secret introduced. Thus, interleaving provides no integrity protection against a MitM attacker.

4.4 Encryption

The interleaved data from the previous step are then encrypted. GSM uses a stream cipher, called A5 in the specification. There are five different variants of A5, called A5/0 to A5/4. These differ in the function that is used for key stream generation. The key stream is then XORed to the plaintext in order to create the ciphertext. Input to keystream generation are a key Kc, which is the result of the authentication procedure, and the frame number. This frame number is incremented every 4.615ms. Output from the key stream generator are 114 bit for upstream and further 114 bit which are used for downstream in the same frame. Using time (or rather, frame number) as input actually makes the attack a bit more complicated for the attacker, as the attacker can't reuse the keystream on retransmissions.

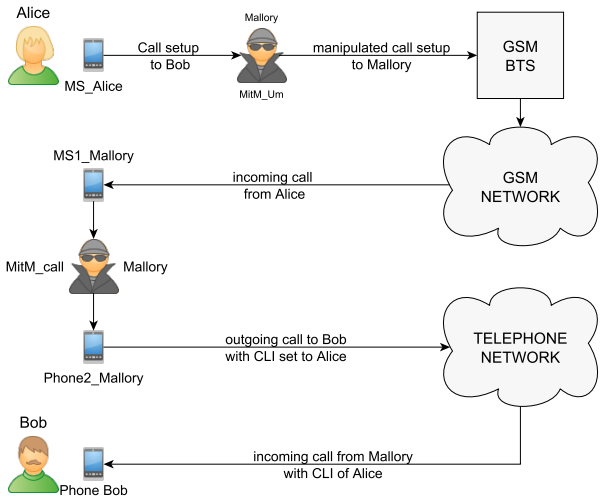

Figure 5: MitM establishment

4.5 Burst Mapping

In addition to two 57 bit blocks, every burst contains two so called *Stealing Flags*. If these are set to one, then the burst is considered to be part of a control channel. In this way, the signaling channel FACCH can use (steal) bandwidth of the voice channel TCH. On dedicated control channels such as SDCCH these stealing flags are always set.

In *Burst Mapping*, the stealing flags are set and two resulting blocks of 58 bit are assigned to bursts. As each block has its own stealing flag, it is possible to assign either of the blocks in a burst to a control channel.

5 THE ATTACK IN DETAIL

5.1 The exploited security flaw

GSM uses a stream cipher without any integrity protection, so from a cryptographic perspective, a MitM-attacker can change all messages undetected and even control the outcome if the plaintext is known in advance. In our attack, the call setup message is modified in order for the attacker to gain access to the plain text of the following call.

5.2 The idea

The attack can be broken into two phases, attack setup and ongoing attack. The starting point is that Alice is registered on a GSM network and tries to call Bob.

To set up the attack, Mallory has installed himself as MitM on the Um interface between Alice's MS and the BTS. Mallory knows Bob's phone number. Thus, Mallory can modify the call setup such that the called number is replaced by Mallory's number (cf. Figure 5). Because there is no integrity protection, neither Alice nor the

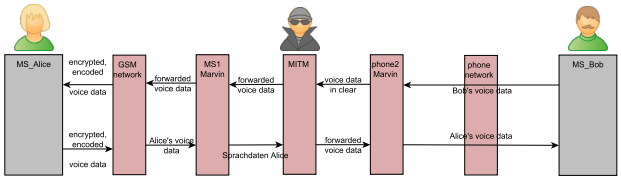

Figure 6: Established MitM-Attack on the speech-connection

network can detect this modification. Thus, the modified call setup will lead to the network setting up the call to Mallory's number. Now Mallory needs to complete the call to Bob, so that Alice doesn't get suspicious. Mallory calls Bob and connects the legs of the call. Note that the second leg doesn't havve to be on the same network. To make sure Bob believes that it is Alice calling him, Mallory can set his calling line identification to Alice's phone number. Customization of Calling Line Identification (CLI) to be presented to the called party is often provided as a legitimate service by telephone operators.

The ongoing attack is quite simple (cf. 6): Mallory doesn't have to remain a man in the middle on the Um interface. The BTS will conveniently decrypt the voice data and forward the content to Mallory, thus Mallory will receive the voice data in clear for the remainder of the call. All Mallory has to do is to forward the voice data received from Alice to Bob and vice versa, while listening in, or even modifying, if so desired.

5.3 Victim and message identification

The fact that Mallory has to blindly modify some bits leaves the questions of how to identify the victim, i.e. Alice, and how to determine the "setup" message in the encrypted data.

For identification of the victim, looking at the mobile originated call establishment, `CM-SERVICE-REQUEST`, which is sent unencrypted, includes the identity of of the originator. This identity is either in the form of the IMSI or the TMSI. To determine which IMSI or TMSI belongs to the victim, it is possible to use an IMSI-catcher, which simply uses a `IDENTITY-REQUEST` to request the IMSI of the MS. According to the specification [33, Subclause 4.3.3.2], the MS has to respond to such a request with its IMSI at any time. To get the mapping of MSISDN to TMSI or IMSI, it is possible to use a cross layer attack and send an Short Message Service (SMS), or even a silent SMS [5], which doesn't even cause a user alert on reception. Sending such an SMS will trigger a paging message to the victim, which is addressed with the victims TMSI.

Now that the attacker knows the victim's TMSI, he filters the SDCCH of the BTS for Set Asynchronous Balanced Mode (SABM)-messages, that piggy back a `CM-SERVICE-REQUEST`. Because they are unencrypted, he can check the field "Connection Management (CM) Service Type" for "CM mobile originated call establishment". All other requested services can be ignored.

Encryption starts with `CIPHERING-MODE-COMMAND`. This is sent unencrypted, and therefore can be identified. From then on, all traffic is encrypted and the attacker can't access layer 2 information. Therefore, the `SETUP` can't be identified explicitly. However, assuming no errors on data link layer, it will always be the same messages sent after `CIPHERING-MODE-COMMAND`: the acknowledgment in LAPDm-Receive Ready (LAPDm) (RR)-frame, and then the `CIPHERING-MODE-COMPLETE`-message. Thus the attacker can determine the `SETUP` message by counting the messages after `CIPHERING-MODE-COMMAND`.

5.4 Message manipulation

Starting from an encoded and encrypted setup message, the manipulation can be represented as follows. First, the necessary variables are defined in Equation 7.

$$\begin{aligned} \mathbb{B} &:= \{0,1\} \\ &\oplus, \text{bitwise exclusive OR (XOR)} \\ \boldsymbol{d} &\in \mathbb{B}^{184}, \text{plaintext data} \\ \boldsymbol{d_m} &\in \mathbb{B}^{184}, \text{manipulated data} \\ \boldsymbol{r'} &\in \mathbb{B}^{228}, \text{blockcode remainder mask} \\ \boldsymbol{k_s} &\in \mathbb{B}^{456}, \text{key stream} \end{aligned} \tag{7}$$

Equation 8 summarizes the functions described in Section 4.

$$\begin{aligned} \boldsymbol{u} &: \mathbb{B}^{184} \to \mathbb{B}^{228}, \boldsymbol{x} \mapsto \boldsymbol{u}(\boldsymbol{x}), \text{block coding} \\ \boldsymbol{c} &: \mathbb{B}^{228} \to \mathbb{B}^{456}, \boldsymbol{x} \mapsto \boldsymbol{c}(\boldsymbol{x}), \text{convolutional coding} \\ \boldsymbol{i} &: \mathbb{B}^{456} \to \mathbb{B}^{456}, \boldsymbol{x} \mapsto \boldsymbol{i}(\boldsymbol{x}), \text{interleaving} \\ \boldsymbol{a} &: \mathbb{B}^{456} \to \mathbb{B}^{456}, \boldsymbol{x} \mapsto \boldsymbol{x} \oplus \boldsymbol{k_s}, \text{ciphering} \\ \boldsymbol{b} &: \mathbb{B}^{456} \to \mathbb{B}^{464}, \boldsymbol{x} \mapsto \boldsymbol{b}(\boldsymbol{x}), \text{burst mapping} \\ \boldsymbol{u_r}(\boldsymbol{x}) &= \boldsymbol{u}(\boldsymbol{x}) \oplus \boldsymbol{r}, \text{block coding with remainder} \\ (\boldsymbol{b} \circ \boldsymbol{a} \circ \boldsymbol{i} \circ \boldsymbol{c} \circ \boldsymbol{u_r})(\boldsymbol{d}), &\text{encoded and enciphered data} \end{aligned} \tag{8}$$

Note that $\boldsymbol{u_r}(0^{184}) = \boldsymbol{r}$.

Convolutional coding, interleaving, ciphering with a stream cipher, and burst mapping are linear.

$$\begin{aligned} \boldsymbol{c}(\boldsymbol{x} \oplus \boldsymbol{y}) &= \boldsymbol{c}(\boldsymbol{x}) \oplus \boldsymbol{c}(\boldsymbol{y}) \\ \boldsymbol{i}(\boldsymbol{x} \oplus \boldsymbol{y}) &= \boldsymbol{i}(\boldsymbol{y}) \oplus \boldsymbol{i}(\boldsymbol{y}) \\ \boldsymbol{a}(\boldsymbol{x} \oplus \boldsymbol{y}) &= \boldsymbol{a}(\boldsymbol{y}) \oplus \boldsymbol{a}(\boldsymbol{y}) \\ \boldsymbol{b}(\boldsymbol{x} \oplus \boldsymbol{y}) &= \boldsymbol{b}(\boldsymbol{x}) \oplus \boldsymbol{b}(\boldsymbol{y}) \end{aligned} \tag{9}$$

Block coding with remainder is affine, thus meaning that it is linear with the remainder XORed to the result of the Fire code without remainder. Thus, the block code of the XOR of two messages is the desired remainder XORed with the XOR of the block code of the two messages:

$$\begin{aligned} \boldsymbol{u_r}(\boldsymbol{x}) \oplus \boldsymbol{u_r}(\boldsymbol{y}) &= \boldsymbol{u}(\boldsymbol{x}) \oplus \boldsymbol{r} \oplus \boldsymbol{u}(\boldsymbol{y}) \oplus \boldsymbol{r} \\ &= \boldsymbol{u}(\boldsymbol{x}) \oplus \boldsymbol{u}(\boldsymbol{y}) \\ &= \boldsymbol{u}(\boldsymbol{x} \oplus \boldsymbol{y}) \end{aligned} \tag{10}$$

$$\begin{aligned} \Rightarrow \boldsymbol{u_r}(\boldsymbol{x} \oplus \boldsymbol{y}) &= \boldsymbol{u}(\boldsymbol{x} \oplus \boldsymbol{y}) \oplus \boldsymbol{r} \\ &= \boldsymbol{u_r}(\boldsymbol{x}) \oplus \boldsymbol{u_r}(\boldsymbol{y}) \oplus \boldsymbol{r} \end{aligned} \tag{11}$$

The attacker would like to modify the message that even though the mobile sent $\boldsymbol{d}$, the base station decodes the message to $\boldsymbol{d_m}$. For the follwing we define the modification of the message as $\boldsymbol{m} = \boldsymbol{d} \oplus \boldsymbol{d_m}$, where $\boldsymbol{m} \in \mathbb{B}^{184}$. With this, Equation 12 holds.

$$\begin{aligned}
&(\boldsymbol{b} \circ \boldsymbol{a} \circ \boldsymbol{i} \circ \boldsymbol{c} \circ \boldsymbol{u_r})(\boldsymbol{d_m}) = \\
&= (\boldsymbol{b} \circ \boldsymbol{a} \circ \boldsymbol{i} \circ \boldsymbol{c} \circ \boldsymbol{u_r})(\boldsymbol{d} \oplus \boldsymbol{m}) \\
&= (\boldsymbol{b} \circ \boldsymbol{a} \circ \boldsymbol{i} \circ \boldsymbol{c})(\boldsymbol{u_r}(\boldsymbol{d} \oplus \boldsymbol{m})) \\
&= (\boldsymbol{b} \circ \boldsymbol{a} \circ \boldsymbol{i} \circ \boldsymbol{c})(\boldsymbol{u_r}(\boldsymbol{d}) \oplus \boldsymbol{u_r}(\boldsymbol{m}) \oplus \boldsymbol{r}) \\
&= \underbrace{(\boldsymbol{b} \circ \boldsymbol{a} \circ \boldsymbol{i} \circ \boldsymbol{c})(\boldsymbol{u_r}(\boldsymbol{d}))}_{\text{Part 1: transmitted bitstream}} \oplus \underbrace{(\boldsymbol{b} \circ \boldsymbol{a} \circ \boldsymbol{i} \circ \boldsymbol{c})(\boldsymbol{u_r}(\boldsymbol{m}) \oplus \boldsymbol{r})}_{\text{Part 2: precompute}}
\end{aligned} \tag{12}$$

Part 1 of the equation above is the bitstream as it is being transmitted. **Part 2** is the XOR pattern that needs to performed on the bitstream over the air. The modification can be precomputed when the message $\boldsymbol{d}$, i.e. the dialed phone number, is guessed, and the manipulation itself can be done bit by bit; there is no need to wait for a complete LAPDm message to arrive.

The stream cipher used in GSM uses a cipher stream that depends on the current frame number. The frame number is synchronized between uplink and downlink, such that the man in the middle attack has to be performed within the same frame. Fortunately for the attacker, GSM offers a feature that allows to time shift the sending of the message from the mobile. The radio layer makes use of the so called timing advance parameter which is used to avoid interference between different mobiles sending at different distances from the base station. Timing advance is used to compensate the propagation delay of the radio signals such that they arrive at the right time at the base station. Thus the delay that a man in the middle attacker introduces will be compensated for by the mobile by increasing its timing advance. The maximal time shift that timing advance can introduce is 63 bit.

GSM is using GMSK for its modulation. This modulation spreads each bit over 11 bits. Because of the current density of base stations, it is rare for a mobile to be as far away from a base station that the full timing advance range is required for normal operation[3]. Thus, an attacker has enough time to demodulate each bit, XOR with the appropriate bit of the XOR pattern $\boldsymbol{v}$ defined as $\boldsymbol{v} = (\boldsymbol{b} \circ \boldsymbol{a} \circ \boldsymbol{i} \circ \boldsymbol{c})(\boldsymbol{u_r}(\boldsymbol{m}) \oplus \boldsymbol{r})$ (cf. Equation 12).

5.5 Countermeasures and Detection possibilities

There are several countermeasures that could be deployed in GSM. The most obvious one would be to include an integrity check value to the control plane traffic. However, this would require updates to both the network and the MS.

The MS could also protect itself without network support. Currently, the easiest would be to avoid usage of GSM. In 3GPP, it was recently standardized that the user can configure the mobile phone to disable its GSM interface. If it is necessary to rely on GSM due to local deployment, the MS could also include a random padding in the call setup message in front of the dialed phone number. The field bearer capability might potentially be used to carry this random padding.

[3]This distance would be 32km.

There is a possibility for network side countermeasures.The network could pick up indicators of the attack: there would be strange timing advance values required to perform the man in the middle attack, or the network could perform transient analysis to do physical device fingerprinting to compare the RF signature of the known device against the detected RF signature. The network could also deliver the called number to the MS after the call, either through USSD or SMS. Tamper protection of that message could be done with an application the SIM card. Alternatively, the network could include a random length padding in this verification message before and after the transmitted phone number, which would make it much harder for an attacker to also attack the downlink message.

6 IMPLEMENTATION AND VERIFICATION

6.1 Virtual physical layer of Um-interface

For verification of the MitM-attack we used the osmoBTS open source GSM system[4]. First, a virtualized physical layer of the Um-interface was implemented. This virtualized physical layer has the advantage of not depending on transceiver hardware, of easy recording of protocol traces, of better turnaround time when testing, and of not requiring a license for use of GSM frequencies.

The implementation was based on the implementation fragment of a virtual BTS within the osmoBTS project. It was possible to reuse functionality that was already implemented in the osmocomBB, osmoBTS, and libosmocore libraries.

To replicate transmission over the air interface, the messages need to contain the time slot and frequency information of the physical layer. In GSM, the frequency is given by the ARFCN, which defines uplink and download frequency. The TDMA time slot defines the physical channel that is being transmitted on. The Frame Number (FN) within the physical channel defines the logical channel. Physical and logical channel determine the destination Service Access Point (SAP), so that the receiver knows how to process the information. No explicit addressing is required. Usually, the Um interface also carries signal quality information. The latter is not important for the virtual interface, as all messages are transmitted with the same signal quality.

To emulate the broadcast properties of the air interface, the virtual air interface uses Internet Protocol (IP) multicast, with one multicast address per frequency and broadcast domain, i.e. different multicast addresses on uplink and on downlink. The transport protocol is User Datagram Protocol (UDP). Within a UDP-packet, a complete LAPDm-frame is encapsulated with the GSM Test Access Protocol (GSMTAP) header that the libosmocore library provides for debugging purposes. GSMTAP has an IANA assigned port number of 4729 and there is a Wireshark[5] dissector available for it.

Choosing GSMTAP for transmission has the great advantage of leveraging existing well tested implementations, but has the drawback of not implementing the actual physical layer processing. In order to show the feasibility of the attack presented in this paper, channel coding and encryption were implemented separately.

For the MS side, the implementation of the physical layer of osmocomBB was replaced by the implementation of the virtual

[4]https://github.com/osmocom/osmo-bts.git
[5]www.wireshark.org

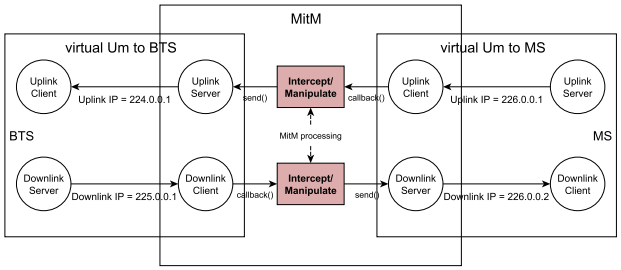

Figure 7: Implementing the MitM in the virtual Um

physical layer. For the BTS side, the same was done for the osmocomBTS project.

6.2 Validation of the attack with virtual Um

To reproduce the capabilities of a MitM in the virtualized environment, the virtual MitM terminates the Um both towards the MS and towards the BTS. This way, the virtual MitM can forward, block or modify messages. This means, that in a real attack, we assume that the attacker sends with higher power towards MS than the BTS and vice versa, thus relegating the actual signal to noise for the receiver. Thanks to automatic power control on the Um-interface, this should be possible.

The implementation of the MitM framework including the virtual Um is available on Github within the osmoMITM repository[6].

To validate the attack, the received LAPDm message is first processed according to the physical layer procedures that the virtual Um interface is skipping. To execute the attack, the MitM module only parses the messages before the ciphering mode command from BTS to MS. After that, the attacker counts the bytes actually on the uplink instead of parsing the LAPDm messages it receives,the attacker predicts which bits to modify by counting the transmitted bytes in the uplink.

7 RELATED WORK – GSM SECURITY AND ATTACKS

7.1 Cryptographic aspects

There are several passive attacks known on confidentiality in GSM, mostly caused by vulnerabilities of the protocols and specification, which are summarized in the following part. Through passive attacks, an attacker can sniff private data, but not manipulate it.

The first cryptanalysis of A5/1, one of the ciphers standardized for GSM, was published three years after its design became known. Golic [15] presented multiple weaknesses and proposed a Time-Memory-Trade-Off attack. Biryukov, Shamir, and Wagner [3] analyzed this attack and found that it was not practical, as it required 15TB of precomputed data and several hours worth of call of known plaintext. However, they extended the attack by exploiting further weaknesses in A5/1 such that 290 GB of precomputed data and approximately 1s of calculation was sufficient. The drawback was that the attack still required the equivalent of two minutes speech of known plaintext. Barkan, Biham and Keller [2] improved this result by exploiting the redundancy on TCH, which is added by error correction mechanisms. This enables the attack to function without known plaintext. Still, several Terabyte of data would have to be precomputed even for of breaking confidentiality of a phone call of only five minutes. At the time of publication, this would have been several CPU years on a commercially available computer, thus limiting the practicality of the attack. At a blackhat conference, Hulton [17] mentioned that calculation of a rainbow table to speed up performing the attack on A5/1 was ongoing. At 26C3, Nohl and Paget [24] presented the first practical attack on A5/1. Their estimate was that calculation of the rainbow table could be shortened from 100000 years to four month by parallelization and execution on 80 Graphics Processing Units (GPUs). At 27C3, Nohl and Munaut [23] actually demonstrated the attack. The rainbow table size was 2TB and precompute time one month with four GPUs. The rainbow tables were made public[7]. The success probability of the attack is estimated at 99% if the registration messages to the network are captured as well, as this gives plenty of known plaintext. Otherwise, the success rate sinks to around 50%. For their demonstration, Nohl and Munaut used two Motorola C123 mobile phones (cost in 2017: approx. 30 Euro) with an adapted firmware from the osmocomBB project.

The algorithm A5/2 was developed for introduction in regions where it was not possible to export strong cryptography into. It was weakened by shortening the used keylength from A5/1. Goldberg, Wagner, and Green [13] published an attack that made it possible to break the cipher in real time. A5/2 was withdrawn from the list of officially supported algorithms by 3rd Generation Partnership Project (3GPP) a few years after Barkan, Biham, and Keller [2] published their practical attack [29].

Dunkelman, Keller and Shamir [8] published a related key attack on the block cipher KASUMI which forms the basis for key stream generation of A5/3. Due to the way KASUMI is used in A5/3, especially because the 64 bit GSM key GSM Cipher Key (Kc) is concatenated to itself for input to the 128 bit key KASUMI algorithm, this attack doesn't translate to an attack against the GSM system. However, 64 bit key length is not considered strong enough, even for legacy systems[8].

To solve the problem of short key length, A5/4 was specified [34] to take 128 bit keys for use with KASUMI. All 3G UMTS Subscriber Identity Module (USIM)-cards will provide a 128 bit key at the end of the authentication run. 2G SIM-cards are not sold by network operators any more, thus in theory 128 bit keys would be available. However, the authors are not aware of any mobile network operator using A5/4 (for the situation in different countries, see e.g. the gsmmap project [16]).

Besides weaknesses in algorithm there is the problem that GSM phones also accept to communicate unencrypted. This isn't much of a problem for passive attacks, as most network operators switch on encryption for all phone calls. However, the standard defines

[6] https://github.com/BastusIII/osmo-mitm.git

[7] https://opensource.srlabs.de/projects/a51-decrypt

[8] cf. https://www.keylength.com, which compiles recommendations from various sources

that phones should be able to indicate the fact that they are communicating unencrypted by presenting a *ciphering indicator* to the user.[9]

Some GSM networks make use of frequency hopping. This poses a problem for passive attacks, even though the main purpose of frequency hopping is to avoid narrow band interference on the radio channel. An attacker trying to mount a passive attack has to know and follow the hopping sequence [23].

7.2 Active attacks over the air interface

Design choices in the authentication and security procedures also allow multiple active attacks, in which the attacker does not only eavesdrop, but also manipulates data. The early GSM-standard only allowed unilateral authentication, the mobile could not check the authenticity of the network it was connected to. Thus an attacker could establish a fake or rogue base station. These rogue BTS were first mentioned in 1996 by Göbel, Leiss, and Marquardt in [12], as IMSI-Catcher. Originally developed by Rohde&Schwarz[10], device "GA 090" was used by authorities to collect the IMSIs identities of mobile subscribers in the vicinity. Collection of IMSI is also possible through another weakness of the GSM-Protocol. Normally, after the very first registration of the mobile in the network, TMSI is used instead of IMSI. However, in case the network loses the mapping from IMSI to TMSI, the network can simply request the mobile to provide its TMSI [33, clause 4.3.3.2]. Such an Identity Request can be sent by an attacker during MS initiated RR-connection establishment. Fox [11] described how small modifications of the IMSI-Catcher software can turn the device into a man in the middle, relaying messages between the MS and the real BTS.

Barkan, Biham, and Keller [2] also proposed using a fake BTS as man in the middle to bid down the phone's security capabilities. The network would then use the weaker algorithm, from which the attacker could derive the key and continue eavesdropping even when the victim starts using a stronger encryption algorithm.

Nohl and Paget [24] presented a concept for an IMSI-catcher built from open source software and affordable hardware. Paget [26] demonstrated this attack in practice, one year later. He simply forced the phones to use A5/0, i.e. Null-encryption. He also demonstrated that it is possible to force phones that would be able to connect using 3G to use 2G, by simply jamming the frequencies used by 3G.

UMTS introduced mutual authentication, which should prevent MitM attacks. Meyer and Wetzel [18] showed that it is still possible to perform a MitM attack against mobile phones that support UMTS and GSM. It works by exploiting a mechanism that was introduced in UMTS to allow backward compatibility with GSM core network infrastructure in a serving network in case of roaming.

Other types of MitM attacks were proposed and demonstrated: by sending binary SMS with Over the Air (OTA) commands, malware could be executed on the USIM-cards of the targeted phone [21]. It is also possible to manage the device remotely, e.g. to set a new Access Point Name (APN) or Hypertext Transfer Protocol (HTTP)-proxy [30], thus permeating the MitM-attack.

[9]The standard also requires that the phone is not presenting the ciphering indicater if the home operator sets a particular flag on the SIM card. Perhaps because the rules are quite complex the ciphering indicator is very rarely implemented.
[10]https://www.rohde-schwarz.com

The importance of integrity protection for encrypted data is well known. Yu, Hartman, and Raeburn [35] demonstrated problems with unauthenticated encryption for Kerberos. Paterson and Yau [27] point to the same conclusion for issues in Internet Protocol Security (IPsec) implementations in Linux. Degabriele and Paterson [7] then implemented and demonstrated these attacks. Most closely related to this work is the work by Bittau, Handley, and Lackey [4] who published an attack on WEP that the key stream from the know protocol header and use this to inject up to 64 byte of traffic. Similar to our attack, Bittau doesn't try to recover the key stream, but manipulate the traffic.

7.3 Other attacks on GSM

Rather than attacking over the air interface, it is possible to attack the network or network elements itself. Golde, Redon, and Borgaonkar [14] attack a femtocell base station and use it for their attacks. A different avenue for attack is to exploit the trust network operators traditionally place in their inter-operator interconnects. Nohl [22] and Mourad [20] show a number of ways how the interconnect network can be misused to gain access to all kinds of information, from call redirection to querying a subscriber's location information.

7.4 Relation to this paper

This work presents the first first MitM attack over the GSM air interface that does not require the cryptographic key and does not require breaking the cryptography. It exploits the lack of integrity protection, similar to the attack on WEP by Bittau et el.[4]. Therefore, it also does not depend on the victim using a weak cipher [2], or being bid down to use a weak authentication [18]. Thus this attack would also work in cases strong encryption algorithms such as A5/4 without the weaknesses described in 7.1. There is no bidding down involved, as in the attacks described in papers about IMSI catchers and protecting against them [1, 6]. Therefore, mutual authentication mechanisms such as UMTS AKA would not prevent this attack.

8 CONCLUSION

This paper presents the feasibility of a man in the middle attack on confidentiality that exploits the lack of integrity protection. The attack essentially turns the base station into a decryption oracle. The work for this paper doesn't actually implement a working exploit in a live network, as we don't yet have the capability of modifying the bits over the air interface. As this paper shows, the security layer has been defeated, the open challenge is to build a repeater that would demodulate the signal and flip predetermined bits. This is left for future work.

The advantage of this attack is that it is independent of the strength of the deployed encryption algorithm, and of the authentication method. It also remains persistent after the initial call setup, even if the attacked MS is not connected to the attacker over Um any more.

The results of this work may translate to other mobile communications systems. In UMTS, the control channel is integrity protected, but it may be possible to attack packet data, as the destination is part of every IP packet. Similar attacks to UMTS may be possible

in Long Term Evolution (LTE). However, it will be more difficult to determine which bits to flip, because the attacker has to determine the beginning of the packets. Furthermore, the attack would be more difficult, as, depending on the type of header compression being used, it may be necessary for the attacker to remain present even after initial setup. A precise analysis of mobile communications generations beyond GSM is left for future work. In the 5G system, that is currently being standardized, integrity protection can also be applied to user plane traffic, thus making it possible thwart this kind of attack completely.

The take away message of this paper is well known: encryption should always be coupled with integrity protection [4, 7, 27, 35], because modification of the ciphertext may break confidentiality.

COORDINATED VULNERABILITY DISCLOSURE

This attack has been reported to the GSMA, an industry interest group of mobile operators worldwide through their coordinated vulnerability disclosure program. At that time, 3GPP did not yet have a coordinated vulnerability disclosure program.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] I. Androulidakis, "Intercepting mobile phone calls and short messages using a GSM tester," in *Computer Networks*, A. Kwiecień, P. Gaj, and P. Stera, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 281–288.

[2] E. Barkan, E. Biham, and N. Keller, "Instant ciphertext-only cryptanalysis of GSM encrypted communication," in *Annual International Cryptology Conference*. Springer, 2003, pp. 600–616.

[3] A. Biryukov, A. Shamir, and D. Wagner, *Real Time Cryptanalysis of A5/1 on a PC*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 1–18. [Online]. Available: http://dx.doi.org/10.1007/3-540-44706-7_1

[4] A. Bittau, M. Handley, and J. Lackey, "The final nail in WEP's coffin," in *Security and Privacy, 2006 IEEE Symposium on*. IEEE, 2006, pp. 15–pp.

[5] N. Croft and M. Olivier, "A silent SMS denial of service (DoS) attack," in *Southern African Telecommunication Networks and Applications Conference 2007 (SATNAC 2007)*.

[6] A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, and E. Weippl, "Imsi-catch me if you can: Imsi-catcher-catchers," in *Proceedings of the 30th Annual Computer Security Applications Conference*, ser. ACSAC '14. New York, NY, USA: ACM, 2014, pp. 246–255. [Online]. Available: http://doi.acm.org/10.1145/2664243.2664272

[7] J. P. Degabriele and K. G. Paterson, "Attacking the ipsec standards in encryption-only configurations," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007, pp. 335–349.

[8] O. Dunkelman, N. Keller, and A. Shamir, "A practical-time attack on the a5/3 cryptosystem used in third generation gsm telephony." *IACR Cryptology ePrint Archive*, vol. 2010, p. 13, 2010.

[9] Ericsson, "Cellular networks for massive iot," no. Uen:284-23-3278, 2016. [Online]. Available: http://www.ericsson.com/res/docs/whitepapers/wp_iot.pdf

[10] P. Fire, *A class of multiple-error-correcting binary codes for non-independent errors*. Department of Electrical Engineering, Stanford University., 1959, vol. 55.

[11] D. Fox, "Der imsi-catcher," *Datenschutz und Datensicherheit*, vol. 26, no. 4, pp. 212–215, 2002.

[12] K. Göbel, L. Leiss, and K. Marquardt, *Strafprozess*. Beck, 1996, p. 46.

[13] I. Goldberg, D. Wagner, and L. Green, "The real-time cryptanalysis of a5/2," *Rump session of Crypto*, vol. 99, pp. 239–255, 1999.

[14] N. Golde, K. Redon, and R. Borgaonkar, "Weaponizing femtocells: The effect of rogue devices on mobile telecommunications." in *NDSS*, 2012.

[15] J. D. Golić, "Cryptanalysis of alleged a5 stream cipher," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1997, pp. 239–255.

[16] gsmmap.org. (2017) Gsm map. [Online]. Available: https://gsmmap.org/

[17] D. Hulton, "Intercepting gsm traffic," *BlackHat Briefings*, 2008.

[18] U. Meyer and S. Wetzel, "A man-in-the-middle attack on umts," in *Proceedings of the 3rd ACM workshop on Wireless security*. ACM, 2004, pp. 90–97.

[19] mobileworldlive.com. (2015, 06) Operators and vendors predict long life for 2g. [Online]. Available: http://www.mobileworldlive.com/featured-content/top-three/operators-vendors-forecast-long-life-2g/

[20] H. Mourad, "The fall of SS7 - how can the critical security controls help," 2015.

[21] K. Nohl, "Rooting SIM cards," *Black Hat USA*, vol. 2013, 2013.

[22] ——, "Mobile self-defense," in *Presentation at Chaos Communication Congress 31C3, Hamburg*, 2014.

[23] K. Nohl and S. Munaut, "Wideband gsm sniffing," in *27th Chaos Communication Congress*, 2010.

[24] K. Nohl and C. Paget, "Gsm: Srsly," in *26th Chaos Communication Congress*, vol. 8, 2009, pp. 11–17.

[25] opensignal.com. (2017) Global cell coverage maps. [Online]. Available: https://opensignal.com/networks/

[26] C. Paget, "Practical cellphone spying," *Def Con*, vol. 18, 2010.

[27] K. G. Paterson and A. K. Yau, "Cryptography in theory and practice: The case of encryption in ipsec," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2006, pp. 12–29.

[28] P. Schnabel, *Kommunikationstechnik-Fibel: Grundlagen, Festnetz, Mobilfunktechnik, Breitbandtechnik, Netzwerktechnik*. Schnabel, 2003. [Online]. Available: http://www.elektronik-kompendium.de/

[29] security.osmocom.org. (2017) Withdrawal of a5/2 algorithim support. [Online]. Available: http://security.osmocom.org/trac/wiki/A52_Withdrawal

[30] M. Solnik and M. Blanchou, "Cellular exploitation on a global scale: The rise and fall of the control protocol," *Black Hat USA*, 2014.

[31] TS-05.02, "Layer 1 Multiplexing and multiple access," 3rd Generation Partnership Project (3GPP), Technical Specification 05.02 - v8.11.0, Jun. 2003. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/0502.htm

[32] TS-05.03, "Channel coding," 3rd Generation Partnership Project (3GPP), Technical Specification 05.03 - v8.9.0, Jan. 2005. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/0503.htm

[33] TS-24.008, "Mobile radio interface layer 3 specification; Core Network Protocols; Stage 3," 3rd Generation Partnership Project (3GPP), Technical Specification 24.008 - v5.15.0, Dec. 2005. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/24008.htm

[34] TS-55.226, "Specification of the A5/4 Encryption Algorithms for GSM and ECSD, and the GEA4 Encryption Algorithm for GPRS," 3rd Generation Partnership Project (3GPP), Technical Specification 55.226 - 10.0.0, Feb. 2011. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/55226.htm

[35] T. Yu, S. Hartman, and K. Raeburn, "The perils of unauthenticated encryption: Kerberos version 4." in *NDSS*, vol. 4, 2004, pp. 4–4.