Department für Informatik
Abteilung Rechnernetze und Telekommunikation

Bachelorarbeit

Sicherheitsanalyse der GSM Luftschnittstelle

Peter Gewald

16. Oktober 2012

1. Gutachter: Prof. Dr. Wolfgang Kowalk
2. Gutachter: Stefan Brunhorn

Ein Großteil der mobilen Kommunikation findet über das globale Mobilfunk-Kommunikationssystem (GSM) statt. Vertrauliche Nachrichten, beispielsweise Transaktionsnummern (TANs) von Banken vertrauen auf die Sicherheit von GSM. Diese Sicherheit basiert hauptsächlich auf einem Verschlüsselungsalgorithmus, der bereits gebrochen ist, jedoch noch großflächig nicht zuletzt von deutschen Netzbetreibern verwendet wird. Durch leichte Modifikation eines Mobiltelefons und der entsprechenden Software ist es möglich einen Angriff erfolgreich durchzuführen. Diese Arbeit zeigt neben der Infrastruktur von GSM die Funktionsweise der Verschlüsselung, theoretische sowie praktische Angriffsmöglichkeiten und wie Provider und Teilnehmer sich davor schützen können. Der praktische Teil konzentriert sich dabei auf das Aufzeichnen und die Entschlüsselung von Kurznachrichten (SMS) mit einem Mobiltelefon. Dabei wird auf aktuelle Projekte, welche sich mit GSM und dessen Sicherheit befassen, hingewiesen, bestehende Software genutzt und teilweise weiterentwickelt.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Motivation

Das *Global System for Mobile Communications* (GSM) ist seit 1992 der dominierende Standard bzgl. Mobilfunkkommunikation in Deutschland. Über GSM werden vertrauliche Gespräche geführt, technische Systeme gesteuert und sogar Bankdaten (z.B. TANs) übertragen. Die Sicherheit der GSM Luftschnittstelle basiert auf Verschlüsselung der Daten mittels eines bestimmten Kryptoalgorithmus. Der in den meisten Industriestaaten am häufigsten verwendete Algorithmus (A5/1) kann mittlerweile in Echtzeit gebrochen werden. Zur praktischen Durchführung solcher Angriffe bedarf es nicht zwangsläufig teurer Hardware. Ein Mobiltelefon in Verbindung mit einem Laptop kann bereits ausreichen, um die Sicherheit von GSM auf die Probe zu stellen. Somit stellt jede vertrauliche bzw. sicherheitsrelevante Kommunikation über GSM ein potentielles Risiko dar. Im Folgenden soll ein einfaches Angriffsszenario beschrieben werden, welches lediglich ein programmierbares Mobiltelefon und entsprechende Software vorraussetzt. Zuvor sollen zum besseren Verständnis neben der GSM Infrastruktur die wichtigsten Kommunikationsabläufe sowie die Funktionsweise der aktuellen Verschlüsselung erläutert werden. Nach der Beschreibung des praktischen Angriffs folgen abschließend Vorschläge zur Verbesserung der sicherheitskritischen Situation, sowohl von den Netzbetreibern als auch von den Teilnehmern ausgehend. Ziel dieser Arbeit ist ein Beitrag zum Schutz der Privatsphäre durch Aufzeigen von leicht ausnutzbaren Schwachstellen innerhalb der mobilen Kommunikation.

2 GSM Infrastruktur

In diesem Kapitel geht es um den logischen und technischen Aufbau eines GSM-Netzes. Beginnend bei den Netzkomponenten werden die Protokolle und die verschiedenen Funkkanäle beschrieben und in Beziehung gesetzt. Insbesondere die Abläufe auf der Funkschnittstelle bilden die Grundlage für die nachfolgenden Kapitel.

2.1 PLNM Infrastruktur

Unter einem *Public Land Mobile Network* (PLMN) versteht man ein landgestütztes, öffentliches Mobilfunknetz. Eines der bekanntesten standardtisierten PLMNs ist das GSM-Netz. Das *Global System for Mobile Comminication* (GSM) ist ein leitungsvermittelndes zellulares Kommunikationsnetz, in dem zwei Teilnehmer über eine explizite (virtuelle) Leitung miteinander kommunizieren. [Sau08, vgl. S.1] [Hei99, vgl. S.14]

2.1.1 GSM Subsysteme

Man unterteilt das GSM-Netz in drei unterschiedliche Subsysteme: [Sau08, vgl. S.12]

BSS Das *Basestation Subsystem* (BSS) beinhaltet Komponenten, welche eine Infrastuktur für die Verbindung zwischen dem Netzwerk (bzw. NSS) und einem mobilen Teilnehmer über die Luftschnittstelle bereit stellen. Wichtige Netzelemente sind die Base Transceiver Station (BTS) sowie der Base Station Controller (BSC) (siehe unten).

NSS Das *Network Subsystem* (NSS) bildet die zentrale Komponente eines GSM-Netzes und ist für die Gesprächsvermittlung und Teilnehmerverwaltung zuständig. Neben anderen mobilen Vermittlungstellen können auch Verbindungen zum nationalen oder internationalen Festnetz aufgebaut werden. Das wichtigste Netzelement stellt das weiter unten beschriebene Mobile Switching Center (MSC) dar.

IN Das *Intelligent Network Subsystem* (IN) beinhaltet keine Unterteilung in Komponenten, sondern besteht aus Datenbanken, welche zusätzliche Dienste zur Verfügung stellen. Zum Beispiel kann über den Prepaid Dienst in Echtzeit Guthaben verwaltet werden.

2.1.2 Komponenten des GSM Netzwerks

Die bereits bei der Definition der Subsysteme angesprochenen Komponenten sollen im Folgenden genauer erklärt werden.

MS Die *Mobile Station* (MS) repräsentiert den Endteilnehmer eines GSM-Netzwerks, meistens in Form eines Mobiltelefons. Der wichtigste Bestandteil einer MS ist die SIM-Karte. Sie enthält u.a. eine geheime Nummer (K_i), welche für die Authentifizierung der MS von Bedeutung ist, jedoch niemals übertragen wird. [Hei99, vgl. S.42] Neben der *Integrated Circuit Card Identification* (ICCID), welche die SIM-Karte eindeutig identifiziert ist jedem Teilnehmer eine eindeutige Nummer zugeordnet, welche ihn innerhalb eines GSM-Netztes weltweit eindeutig identifiziert. Diese sogenannte *International Mobile Subscriber Identity* (IMSI) besteht aus drei Bestandteilen. Die ersten drei Ziffern bilden den *Mobile Country Code* (MCC), welcher das zugehörige Land identifiziert (z.B. 262 für Deutschland). Die nächsten zwei (selten auch drei) Ziffern bilden den *Mobile Netwotk Code* (MNC), der den Provider identifiziert (z.B. 01 für T-Mobile). Über die Vergabe des zugehörigen MNC an den entsprechenden Betreiber entscheidet in Deutschland die Bundesnetzagentur. Auf Internationaler Ebene entscheidet die *International Telecommunication Union* (ITU), welche für die Spezifikation der IMSI verantwortlich ist. [1] Die letzten zehn Ziffern der IMSI sind dem Teilnehmer von dem jeweiligen Provider individuell zugeordnet und werden als *Mobile Subscription Identification Number* (MSIN) bezeichnet.

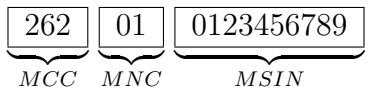

Abbildung 2.1: Aufbau der IMSI

Neben der IMSI ist der MS noch eine Rufnummer, die *Mobile Subscriber ISDN Number* (MSISDN) zugeteilt. Einer IMSI können mehrere Rufnummern zugeordnet werden, da die IMSI (und nicht die MSISDN) als Primärschlüssel fungiert. So

[1] http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-E.212-200805-I!!PDF-E&type=items

kann die Rufnummer jederzeit geändert werden ohne die SIM-Karte auszutauschen, da die MSISDN nicht fest auf der SIM-Karte gespeichert ist. Die MSISDN besteht aus einem Country Code (CC), z.B. 49 für Deutschland und einer dreistelligen Vorwahl des Netzbetreibers, dem National Destination Code (NDC), beispielsweise (0)162[2]. [Sau08, vgl. S.20]

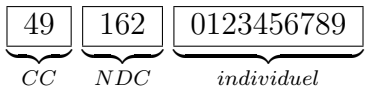

Abbildung 2.2: Aufbau der Rufnummer (MSISDN)

BTS Die *Base Trancievер Station* (BTS) bildet die Schnittstelle zwischen der kabelgebundenen Verbindung und der Luftschnittstelle. Ein BTS kann theoretisch „eine Fläche mit einem Radius von bis zu 35 km abdecken." [Sau08, Zitat S.31] Bei den meisten BTSs, inbesondere in städtischen Gebieten beträgt der Radius aber nur 3 bis 4 km, in Wohngebieten sogar nur mehrere 100 m. [Sau08, vgl. S.32] Diese abgedeckte Fläche nennt man auch *Zelle*. Während einer Verbindung werden die Gesprächsdaten über eine BTS geschickt, welche auch im Laufe der Verbindung wechseln kann. Die Sendeleistung und damit auch die Reichweite kann je nach Wohndichte abweichen, wobei das Endgerät normalerweise die begrenzende Komponente ist. Die Zellen haben eine eindeutige ID (*Cell-ID*) und werden meist in Waben modelliert (siehe Abbildung 2.3).

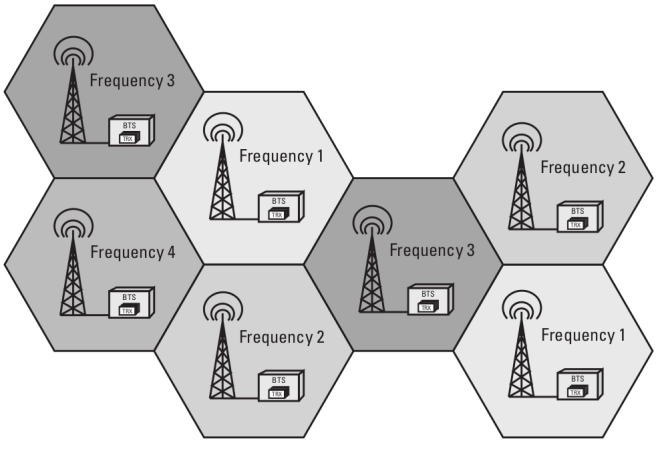

Abbildung 2.3: Warbenstrukter eines GSM Netzes aus [Hei99, S.4]

[2]Früher war diese Nummer Vodafone eindeutig zugeordnet: http://www.postsitter.de/auskunft/handyvorwahlen.htm (Mittlerweile kann die Rufnummer auch auf andere Provider portiert werden)

Mehrere Zellen (ca. 20) [Sau08, S.64] werden zu einer *Location Area* (LA) zusammegefasst. Eine Location Area ist über eine eindeutige Nummer, der *Location Area Identity* (LAI) definiert, welche sich aus der oben genannten MCC, MNC und einer individuellen Kennung, dem *Location Area Code* (LAC) zusammensetzt. Zu den Aufgaben der BTS gehören u.a. die Verwaltung der Funkkanäle, Modulation der Signale und Ver- und Entschlüsselung der Daten. Die Antennen decken lediglich ein Bereich von maximal 180° ab, weshalb an einem Funkturm meistens mehrere Antennen montiert sind, oftmals drei Antennen mit einer Abdeckung von je 120° (siehe Abbildung 2.4). Allein in Deutschland gab es 2001 ca. 52000 Basisstationen. [3]

Abbildung 2.4: Antennen eines BTS-Turms

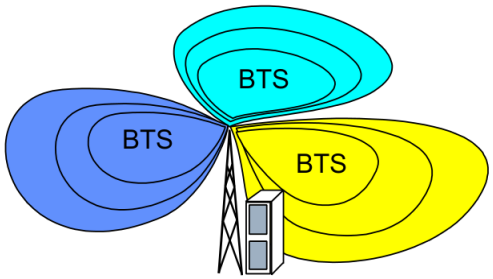

Abbildung 2.5: BTS-Turm mit drei 120° Antennen aus [Oy02, S.24]

BSC Der *Base Station Controller* (BSC) bildet die zentrale Einheit des BSS und bündelt die Verbindungen aller an ihn angeschlossenen BTSs. Die Verbindungsverwaltung wird über eine Schaltmatrix realisiert, welche Daten von den einzelnen BTSs zu einem MSC (siehe unten) und umgekehrt rangiert. Die sogenannte A-bis

[3] http://www.karguti.de/Arbeiten/Vertiefung/5_VT_Schaedlichkeit_Stumpp_1.pdf

Schnittstelle (BTS ↔ BSC) wird meist mittels einer 2 Mbit Leitung realisiert, die sich in virtuelle Kanäle mit einer Bandbreite von 64 kBit/s gliedert. Zu den Aufgaben des BSC gehören neben der Steuerung der BTS (z.B. durch Leistungsregelung) vor allem die Organisation des Zellwechsels (*Handover*). Ein Handover ist nur Aufgabe des BSC, solange sich die Quell- und Ziel BTS innerhalb des selben BSS befinden. Befinden sich beide Funktürme in unterschiedlichen BSS, so wird die Verwaltungsverantwortung für den Handover an das MSC übergeben. Dem BSC obligt bei einer schwachen Signalstärke einer MS die Entscheidung in welche Zelle ein Handover erfolgen soll. [Sau08, vgl. S.45] Der BSC verfügt des Weiteren über eine Datenbank, in der Statusinformationen über die gesamte BSS gespeichert sind, wie z.B. sämtliche Zellen seiner Location Area und sämtliche Signalstärken der Teilnehmer. Diese Komponente wurde eingeführt, um das MSC zu entlasten, sodass sich der Netzaufbau vom MSC bis zum Endteilnehmer als Baumtopologie darstellen lässt. [Sau08, vgl. S43ff] [Hei99, vgl. S.36f]

MSC Das *Mobile Switching Center* (MSC) bildet die Hauptkomponente eines NSS, welche mehrere BSS verwaltet und kontrolliert. Die Gebührenabrechnung und Authorisierung von Mobilstationen sowie die damit verbundene Aufzeichnung von Gesprächsdaten gehören zu den wichtigsten Aufgaben dieser Komponente. Sobald eine MS eingeschaltet wird, registriert sie sich beim MSC und ist für andere Teilnehmer erreichbar. Der Aufbau einer Verbindung sowie das Weiterleiten von SMS[4] wird von dieser Komponente kontrolliert. [Sau08, vgl. S.13] Wechselt eine MS die Location Area, muss sie ein *Location Area Update* durchführen, also dem zuständigen MSC mitteilen, in welcher Location Area sie sich befindet. Das MSC kann die MS somit bei einer ankommenden Verbindung innhalb dieser Location Area suchen, indem sie eine Paging-Anfrage an alle Zellen (BTS) schickt, welche innerhalb dieser Location Area liegen. [Sau08, vgl. S.63f] Ein großes Mobilfunknetzwerk besteht „normalerweise aus dutzenden oder sogar hunderten voneinander unabhängigen MSCs.“ [Sau08, Zitat S.15] Für die Weiterleitung und Speicherung von Kurznachrichten (SMS) existiert neben den MSCs noch das *Short Message Service Center* (SMSC), welches sich um den Versand und die Verwaltung von SMS kümmert. [Sau08, S.26f] Das MSC verwendet zum Verwalten von Mobilteilnehmern Datenbanken die im Folgenden beschrieben werden.

HLR Das *Home Location Register* (HLR) repräsentiert die wichtigste Datenbank innerhalb eines GSM-Netzes. Es ist ein statisches Register, welches jeden Teilnehmer an-

[4]Short Message Services (Kurznachrichten)

hand der IMSI als Primärschlüssel fest zuordnet. Das HLR behinhaltet [Sau08, vgl. S.21ff]:

- IMSI
- Basisdienste (z.B. Telefonie, SMS, Datendienste, FAX)
- Zusätzliche Dienste (z.B. CLIR[5])
- Zuordnung IMSI $\leftrightarrow$ Rufnummer (MSISDN)
- *Authentication Center* (AC) welches Authentifizierungsinformationen (z.B. den geheimen Schlüssel K_i der SIM-Karte) speichert.
- MSRN (Mobile station roaming number) ist eine temporäre ID zum Auffinden von Teilnehmern fremder Netze.

VLR Jedes MSC verfügt über eine eigene Datenbank, welche einen Auszug aus dem HLR enthält. [Sau08, vgl. S.17] Dieses sogenannte *Visistor Location Register* (VLR) kann man sich als temporäres HLR vorstellen, dass die Teilnehmer beinhaltet, welche an einer „MSC-eigenen“ BTS angemeldet sind. Bei der Prüfung, ob sich eine Mobilstation im Einflussbereich einer MSC befindet, muss nun nicht mehr das globale HLR angefragt werden, sondern lediglich das lokale VLR. Dies hat einen performanteren Verbindunsaufbau und weniger Belastung des HLRs zur Folge.
Während das HLR eher statische Informationen speichert, behinhaltet das VLR überwiegend dynamische Daten: [Hei99]

- IMSI
- TMSI
- Rufnummer (MSISDN)
- LAI (= MCC + MNC + LAC)
- MSRN
- Handover Nummer

Die *Temporary Mobile Subscriber Identity* (TMSI) ist eine temporäre ID, welche anstelle der IMSI übertragen wird. Damit soll die Anonymität eines Teilnehmer sichergestellt und Bewegungsprofile vermieden werden.

EIR Das *Equipment Identity Register* (EIR) verwaltet die Hardwareidentifikationsnummern der Mobilstationen, die sog. *International Mobile Equipment Identity* (IMEI).

[5] Calling Line Identification Restriction (Rufnummerunterdrückung)

Jedes Mobiltelefon verfügt über eine solche weltweit eindeutige Hardwarenummer. Die IMEI besteht aus einem Type Allocation Code (TAC), welche neben der Zulassungsstelle (erste zwei Ziffern) den Gerätetyp identifiziert [6]. Nach dem TAC folgt eine individuelle Seriennummer (SNR) und abschließend noch eine Checksumme[7] (siehe Abbildung 2.6[8]).

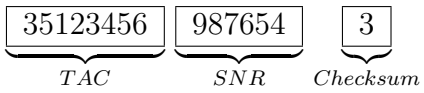

Abbildung 2.6: Aufbau der IMEI ab 2004

Das EIR teilt sich in drei verschiedene Listen. Die „weiße“ Liste besteht aus allen vorhandenen IMEIs, die „graue“ Liste entält IMEIs, die zur Verfolgung freigegeben sind und die „schwarze“ Liste beinhaltet die IMEIs gestohlener bzw. gesperrter Mobiltelefone. Gestohlende Mobiltelefone können über die schwarze Liste abgefragt und ggf. lokalisiert werden, sodass ein Austauschen der SIM-Karte für eine Verschleierung des Diebstahls nicht ausreichend ist. Eine Änderung der IMEI ist softwareseitig schwer möglich, sie ist jedoch gegenüber dem Netz emulierbar. [9] Das EIR ist eine zusätzliche Datenbank, deren Einsatz providerabhängig ist. [Hei99, vgl. S.7, 38]

Aus den oben beschriebenen Komponenten ergibt sich die Architektur eines GSM Netzes (Abbildung 2.7).

2.2 Air-Interface

Das Air-Interface (bezogen auf GSM spricht man vom Um-interface) bezeichnet die Luftschnittstelle zwischen einer Mobilstation und der BTS.

2.2.1 Protokolle

Das Air-Interface von GSM gliedert sich in drei Schichten, [10] welche im Folgenden erläutert werden. Dieser Abschnitt soll einen Überblick über den strukturellen Aufbau der GSM

[6] Identifikationen z.B. möglich via: https://www.numberingplans.com/?page=analysis&sub=imeinr
[7] Ein Generator zum Berechnen der Checksumme nach dem Luhn-Algorithmus ist hier zu finden: http://imei.sms.eu.sk/
[8] vgl. http://imei-number.com/imei-structure/
[9] http://www.shop-alarm.de/Abhoersicheres_Handy_Mobiltelefon_-_Aendern_der_IMEI_GSM_abhoersicher_Seriennummer_aendern.html
[10] Die Schichten beziehen sich auf das OSI-7-Schichtenmodell

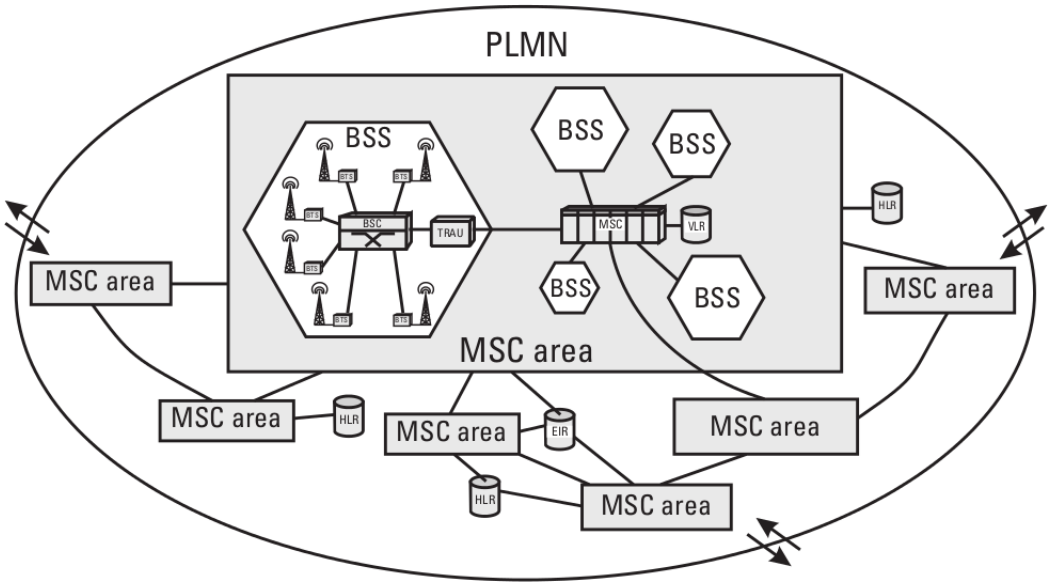

Abbildung 2.7: Architektur eines GSM Netzes aus [Hei99, S.5]

Luftschnittstelle geben und zeigen, an welcher Stelle potentiell gesuchte Informationen zu finden sind.

Schicht 1

Auf der physikalischen Schicht des Um-interfaces werden zwei verschiedene Multiplexing-Techniken eingesetzt, um möglichst vielen Teilnehmern gleichzeitige Kommunikation zu ermöglichen. Das erste Verfahren teilt die Frequenz pro Zelle auf mehrere Teilnehmer auf und wird daher als *Frequency Division Multiple Access* (FDMA) bezeichnet. Die zweite Technik teilt den Kanal in mehrere Zeitschlitze (*Bursts*) auf und trägt die Bezeichnung *Time Division Multiple Access* (TDMA). Pro Trägerfrequenz mit 200 kHz Bandbreite können so mit Hilfe von TDMA bis zu 8 Teilnehmer gleichzeitig kommunizieren. Ein TDMA Frame besteht demnach aus 8 Zeitschlitzen und wird in 4,615 ms übertragen. Unter der Annahme, dass eine BTS auf n verschiedenen Frequenzen funkt, ergäben sich abzüglich 2 Kanäle für die Signalisierung auf der ersten Trägerfrequenz $8n-2$ Zeitschlitze. [Sau08, vgl. S.33f]
Ein GSM-Burst (siehe Abbildung 2.8) besteht zunächst aus einer *guard time*, welche einen Zeitpuffer darstellt, in der keine Daten übertragen werden. Dies liegt in der Mobilität der Teilnehmer begründet, die unterschiedliche Entfernungen zur BTS haben und somit Daten zeitverzögert ankommen können. Damit der Empfänger den Anfang und das Ende eines Burst identifizieren kann, wird ein bekanntes Bitmuster (drei Nullen), *tail* genannt, mitgeschickt. In der Mitte des Burst befindet sich ein immer gleiches Bitmuster, welches

der Fehlerkorrektur dient. Mit Hilfe dieses Musters (auch *training sequence* genannt) können Signalverfälschungen, welche z.B. durch Reflexion/Mehrfachausbreitung entstehen, ausgeglichen werden. Schließlich gibt es noch sog. *stealing flags*, welche angeben, ob sich in den Datenfeldern Nutzdaten oder Signalisierungsinformationen befinden. Jeder GSM-Burst dauert 577 µs und überträgt 114 Bit Nutzdaten über die unten genannten logischen Kanäle. [Sau08, vgl. S.34f]

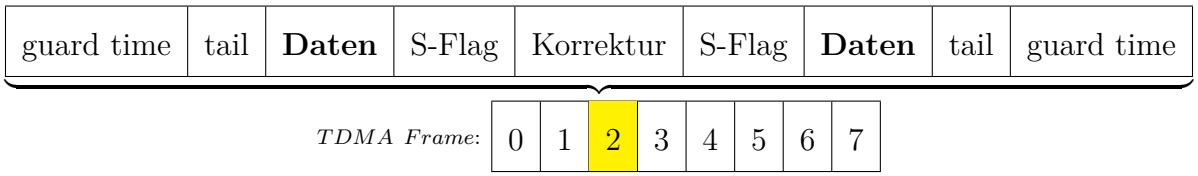

Abbildung 2.8: Ein GSM-Burst nach [Sau08, S.35]

Für die Synchronisation, Frequenzkorrektur und Access haben die Bursts ein anderes Format als in der obigen Abbildung 2.8 gezeigt ist. [Kow02, vgl. Funksubsystem] Da es mehr Signalisierungskanäle als Zeitschlitze gibt, sind im GSM 51 TDMA-Frames zu einem sich ständig wiederholenden Multiframe zusammengefasst, in dem festgelegt ist, in welchem Burst (Zeitschlitz 0 und 1 innerhalb eines TDMA-Frames) welche logischen Kanäle übertragen werden. [Sau08, vgl. S.36] Neben dieser gängigen Konfiguration ist es auch möglich, lediglich den ersten Zeitschlitz für die Signalisierungskanäle zugunsten der Nutzdatenkanäle zu verwenden. [Hei99, vgl. S.108] Die Nutzdatenkanäle sind in einem sich wiederholenden 26er Multiframe angeordnet, der parallel zum 51er Multiframe verläuft. Die obige Aufteilung gilt nur für Frames auf der ersten Trägerfrequenz. Alle weiteren

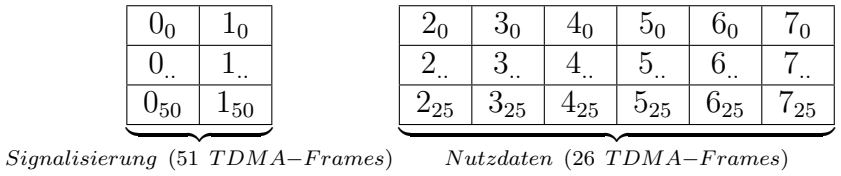

Abbildung 2.9: Nutzung der Timeslots innerhalb der zwei Multiframes

Frequenzen beinhalten lediglich 26er Multiframes und können somit alle 8 Burst für Nutzdaten verwenden.

Schicht 2

Auf der Sicherungsschicht der Luftschnittstelle wird das $LAPD_m$ Protokoll verwendet. Es handelt sich dabei um eine Abwandlung des *Link Access Procedure for the D-Channel*

(LAPD) des ISDN D-Kanals. Das m steht dabei für *modified*. [Hei99, vgl. S.101] Es gibt drei verschiedene Formate des 184 Bit langen $LAPD_m$-Frames. Das A-Format wird als „Lückenfüller"-Frame verwendet, sofern keine zu sendenen Nutzdaten vorliegen. Das B-Format wird zum Transport von SDCCHs, FACCH und SACCH, wobei die Maximallänge der Nutzdaten vom Channel abhängt. Schließlich gibt es noch das Bbis-Format, welches der Übertragung von BCCH, PCH und AGCH, also lediglich dem Downlink dient. Da die Nachrichten dieser Channels an alle Teilnehmer einer Location Area geschickt werden, gibt es keine Identifikationsnummern bei Bbis-Frames. Im Folgenden soll der Aufbau eines $LADP_m$-Frames am Beispiel des B-Formats gezeigt werden, in dem sich beispielsweise ein SDCCH befinden kann, der z.B. eine SMS transportiert. Ein B-Frame besteht zunächst aus einem Feld, das den Frame auf 184 Bit auffüllt, da einige Channels weniger Signaldaten transportieren als andere. Das Feld, welches die Framelänge (inklusive Segmentierungsinformationen) der Signaldaten beinhaltet und die letzten beiden Felder haben je eine Länge von 8 Bit. Das Kontrollfeld beschreibt den Frametyp und gibt an, ob es sich um einen Informationsframe, Überwachungsframe (z.B. zur Angabe der Empfangbereitschaft) oder unnummerierten Frame (z.B. Verbindunggsaufbau/-abbau) handelt. Framechecksummen wie im LAPD für ISDN sind nicht vorhanden, da sich die erste Schicht bereits um Fehlerkorrekturen kümmert.

Auffüllen	**Signaldaten**	Framelänge	Kontrollfeld	*Adressfeld*

Abbildung 2.10: Ein $LADP_m$ B-Frame nach [Hei99, S.103]

Das Adressfeld[11] (Abbildung 2.11) gibt genauere Informationen über den Frame und sei im Folgenden etwas detaillierter betrachtet. Der 2 Bit lange Long Link Protocol Discriminator (LPD) codiert den Cell Broadcast Service (CBS) mit 01 und ist ansonsten immer 00. Der Service Access Point Identifier (SAPI) (3 Bit) klassifiziert die Signaldaten (z.B. 0 für CC/MM/RR (siehe unten) oder 3 für Zusatzdienste (SS) und SMS), während das Command or Response (C/R) Feld eine 0 für Kommandoframes und eine 1 für Antwortframes kodiert. Schließlich makiert das Extended Address (EA) Feld, ob noch ein weiteres Adressoktett folgt und hat im $LDAP_m$ daher immer den Wert 1, da das Adressfeld das letzte Adressoktett ist. [Gö03, vgl. S.27] [Hei99, vgl. S.102ff]

[11] Ergänzende Informationen aus der ISDN-Welt (LAPD):
http://networking.ringofsaturn.com/Telecommunications/isdn.php

(frei)	LPD	SAPI	C/R	EA

Abbildung 2.11: Adressfeld eines $LADP_m$ Frames nach [Hei99, S.103]

Schicht 3

Die 3. Schicht der Luftschnittstelle unterscheidet Nachrichtentypen, welche wiederum im Gruppen eingeteilt sind. Die Typ ID beinhaltet den Protocol Diskriminator (PD), welcher die Nachrichten bzw. Meldungen in Gruppen einteilt. Mögliche Gruppen sind:

- Radio Resource Management (RR) mit PD[12] = 6
- Mobility Management (MM) mit PD = 5
- Call Control (CC) mit PD = 3
- Rufunabhängige Supplementary Services (SS) mit PD = B
- SMS Meldungen mit PD = 9

Jeder Gruppe sind verschiedene Meldungstypen zugeordnet. So gehört z.B. der Meldungstyp *Immediate Assignment* zur RR-Gruppe, Meldungstypen bezüglich eines Location Updates der MM-Gruppe und Meldungen für die Rufsteuerung (z.B. *Connect, Disconnect*) der CC-Gruppe. [Gö03, S.30ff]

Typ ID (PD)	Nachrichtentyp	**Daten**

Abbildung 2.12: Vereinfachter Schicht 3 Frame [Hei99, vgl. S.107]

Die ersten zehn Meldungen für das Radio Resource Management (RR) sind in Tabelle 2.1 aufgelistet, da sie in Bezug auf das Zustandekommen einer Nutzdatenverbindung am wichtigsten sind. Zu den RR-Meldungen gehören noch *Classmark Enquiry, Measurement Report, Ciphering Mode Command, Ciphering Mode Copmlete, Assignment Command, Assignment Complete, Channel Release* und *Handover*. [Gö03, S.30ff]

2.2.2 Channelübersicht

Die Kommunikation zwischen der Mobilstation und der BTS teilt sich in diverse logische Kanäle (Channels), die im Folgenden beschrieben werden. [Kow02, vgl. Funksubsystem] In welcher Kombination die einzelnen Channels innerhalb eines Multiframes auftreten,

[12] Hier in hexadezimaler Schreibweise, also $6_{16} = 0110_2$

RR-Meldung	Beschreibung
Paging Request 1	Paging Anfrage an alle BTS innerhalb einer Location Area. Der Empfänger eines Gespächs oder einer SMS wird gesucht.
System Information Type 1	Die BTS teilt mit, über welche Kanäle sie verfügt (Cell Allocation). Ein Kanal identifiziert sich über die ARFCN (Absolute Radio Frequency Channel Number, Up- und Downlinkfrequenzen).
System Information Type 2	Die Frequenzen der Nachbarzellen (Broadcastkanäle) einer BTS werden übergeben.
System Information Type 3	Informationen über die Zelle, z.B. Cell-ID, LAC und Parameter bzgl. verfügbarer Kanäle
System Information Type 4	Enthält zum Teil Informationen der vorigen Meldungen und Kanaleigenschaften bzgl. GPRS
System Information Type 5	Kanäle, in die sich die MS während eines Gesprächs einbuchen kann.
System Information Type 6	Der MS wird mitgeteilt wo sie sich befindent (Cell-ID und LAC)
Channel Request	Kanalanforderung einer MS inkl. Grund, z.B. als Antwort auf eine Paging Request, (Not)ruf oder Location Update
Immediate Assignment	Beantwortung einer Channel Request und Zuweisung eines speziellen Nutzdatenkanals (siehe SDCCH in der Channelübersicht)
Paging Response	Aushandlung von Parametern zum Gesprächsaufbau, z.B. welchen Verschlüsselungsalgorithmus MS und MTS unterstützen, Leistung der MS und mögliche Frequenzbänder der MS.

Tabelle 2.1: Wichtige RR-Meldungen nach [Gö03]

variiert und hängt von Netzbetreiber ab. Der Uplink definiert sich über die Richtung $MS \rightarrow BTS$ und der Downlink folglich über die Richtung $MS \leftarrow BTS$. [Hei99, vgl. S.105]

Broadcast Channels

Die *Broadcast Channels* (BCH) verwendet die BTS für Punkt-zu-Mehrpunkt-Kanäle zu den jeweiligen MSs. Bei diesem Channels handelt es sich also um Downlink Channels.

FCCH Der *Frequency Correction Channel* (FCCH) enthält sog. Freuquency Correction Bursts, ein unmoduliertes Signal, welches zur Synchronisation (Anfang des 51-Multiframes finden) und später der Fehlerkorrektur der Freuquenz zwischen MS und BTS dient. [Sau08, vgl. S.38]

SCH Mit Hilfe des *Synchronization Channels* (SCH) kann die MS eine Zelle bzw. den BCCH einer BTS finden und sich mit dieser synchronisieren.

BCCH Der *Broadcast Control Channel* (BCCH) dient als Identifikationskanal einer Zelle. Über diesen Channel werden u.a. der Location Area Code (LAC), MCC, MNC, Cell ID, die Up- und Downlinkfrequenzen (ARFCN) sowie die Frequenzen der Nachbarzellen bekannt gegeben. Die Rahmennummer, mit der eine Anfrage auf dem PCH (siehe unten) gesendet wird oder welche Rahmennummern für eine Anfrage auf dem RACH verfügbar ist, gehört ebenfalls zum Informationsspektrum des BCCH. [Hei99, vgl. S.105] [Sau08, vgl. S.38]

CBCH Der *Cell Broadcast Channel* (CBCH) wird verwendet, um spezielle Informationen (z.B. Nachrichten, Wetter, Verkehr) an alle Teilnehmer in Reichweite zu verteilen. Der CBCH ist zwar logisch gesehen ein Broadcast Channel, aus technischer Sicht jedoch dem SDCCH (siehe unten) zugeordnet, innerhalb dessen Zeitschlitz er auch sendet.[13] Die von diesem Channel gesendeten Botschaften werden nicht vom Mobiltelefon bestätigt.

Common Control Channels

Die *Common Control Channels* (CCCH) dienen vorwiegend dem Verbindungsaufbau und betreffen meistens mehrere Teilnehmer.

PCH Der *Paging Channel* (PCH) benarichtigt („ruft“) die MS über eingehende Daten, z.B. einen Anruf oder eine SMS. Eine Paging-Anfrage wird an alle Zellen einer Location Area geschickt, in der sich der zu rufende Teilnehmer befindet. Als Rufname wird die IMSI bzw. TMSI verwendet. [Sau08, vgl. S.40]

RACH Über den *Random Access Channel* (RACH) kann eine Mobilstation Kommunikationsanfragen an die BTS schicken. Diese Anfrage beinhaltet meistens den Zugriffswunsch eines speziellen (dedicated) Channels. Dazu sendet die MS eine Channel Request Nachricht. Da die verschiedenen Teilnehmer nicht untereinander synchronisiert sind, können nicht regelbare Kollisionen auftreten. Der RACH ist damit ein reiner Uplink-Channel. [Sau08, vgl. S.40]

AGCH Der *Access Grant Channel* (AGCH) informiert die MS nach erfolgreicher Kommunikation auf dem RACH mit einer Immediate Assignment Nachricht über eine

[13] Dieses Vorgehen wird auch als „Frame Stealing“ bezeichnet.

initiale Zuweisung an einen SDCCH. Über die Vergabe der SDCCHs und TCHs entscheidet der BSC. [Sau08, vgl. S.40f]

Dedicated Control Channels

Die *Dedicated Control Channels* (DCCH) sind spezielle Channels, welche lediglich einen Teilnehmer betreffen. Bis auf den TCH sind sie mit dem Datenkanal eines ISDNs vergleichbar. Alle DCCHs sind sowohl Uplink als auch Downlink-Channel ($MS \leftrightarrow BTS$).

SDCCH Der *Standalone Dedicated Control Channel* (SDCCH) wird für die Signalisierung und den initialen Aufbau von Anrufen zwischen der MS und der BTS verwendet, wenn dem Teilnehmer also noch kein TCH (siehe unten) zur Verfügung steht. Darüber hinaus beinhaltet der SDCCH Signalisierungsdaten, welche keinen TCH benötigen, wie z.B. ein Location Update oder das Senden bzw. Empfangen einer SMS. [Sau08, vgl. S.38]

FACCH Der *Fast Associated Control Channel* (FACC) ist ein Steuerungskanal, welcher dringende Signaldaten, wie z.B. ein Handover Kommando während einer bestehenden Verbindung überträgt. Da solche dringenden Signalisierungsnachrichten vergleichsweise selten vorkommen, wird dem FACCH kein eigener Burst zur Verfügung gestellt. Statt dessen wird er anstelle der Nutzdaten übertragen und im GSM Burst das stealing flag gesetzt. Er wird in dem Zeitschlitz des TCH (siehe unten) ausgeführt und ist aus technischer Sicht ihm zugehörig. [Sau08, vgl. S.36]

SACCH Der *Slow Associated Control Channel* (SACCH) beinhaltet im Uplink Signalmessungen der aktiven Zelle und den Nachbarzellen. Aufgrund dieser Ergebnisse kann beispielsweise ein Handover oder eine Leistungsanpassung angeordnet werden. Zweiteres wird im Downlink des SACCHs zusammen mit Timing-Informationen mitgeteilt. Da diese Informationen eine aktive Verbindung betreffen, wird der SACCH zusammen mit dem TCH aufgebaut. [Sau08, vgl. S.37]

TCH Der *Traffic Channel* (TCH) ist der Sprach bzw. Nutzdatenkanal und entspricht dem Basis Kanal eines ISDNs. GSM verwendet unterschiedliche Typen des TCH. Man unterscheidet zunächst den Fullrate-speech traffic channel (TCH/FS) und den Halfrate-speech traffic channel (TCH/HS). Darauf aufbauend können TCHs unterschiedlicher Übertragungsrate definiert werden, z.B. TCH/F9.6 für einen TCH/F mit 9.6 Kb/s. [Hei99, S.402, Table G.18]

Frequencyhopping

wird innerhalb eines DCCHs eingesetzt, um Störungen zwischen benachbarten Zellen zu verhindern. Um Frequencyhopping zu betreiben benötigt man mindestens eine der folgenden Informationen [Hei99, vgl. S.353f]:

Cell allocation (CA) Eine Liste aller verfügbaren Frequenzen (ARFCNs) innerhalb der eigenen Zelle (von BCCH gesendet)

Mobile allocation (MA) Eine Auswahl von Frequenzen von der CA-Liste zusammen mit der hopping Sequenz.

Hopping sequence number (HSN) Ein Wert zwischen 0 und 63, der die Sprungweite angibt

Mobile allocation index offset (MAIO) Frequenznummer, die ebenfalls zwischen 0 und 63 (Wertebereich von MA) liegt. Anhand dieses Offsets werden die MS innerhalb eines TDMA-Frames über alle verfügbaren Frequenzen verteilt.

Frame numer (FN) Variabler Wert, der Zähler beinhaltet, welche die hopping Sequenz ändern.

Bei einem Immediate Assignment auf dem AGCH werden auch MAIO und HSN mit dem neu zugewiesenen SDCCH mitgeschickt. Frequencyhopping verfolgt während einer Sprachverbindung auch die Intention die Verbindung vor Angriffen zu sichern. Es gilt als erwiesen, dass es lediglich etwas mehr Berechnungen und Bandbreite bedarf, um diesen Schutz zu umgehen. [Dia12]

2.3 Kommunikationsbeispiele

Die folgenden Beispiele sollen die oben genannten Channels und GSM-Komponenten im Einsatz ausgewählter Szenarien zeigen. Konkrete Verschlüsselungs- und Authentisierungsaspekte sind bei folgenden Szenarien noch nicht berücksichtigt.

2.3.1 Aufbau eines Sprachkanals

Das folgende Szenario beschreibt die Etablierung eines Sprachkanals auf der Seite des Anrufempfängers.
Der BSC empfängt vom MSC eine Paging-Anfrage mit IMSI, TMSI und Location Area des Empfängers. Daraufhin sendet der BSC eine Paging-Anfrage an alle BTS innerhalb der

angegebenen Location Area, welche er seiner Datenbank entnimmt. Sämtliche BTS leiten die Paging-Anfrage über den PCH an alle sich in Reichweite befindlichen Teilnehmer weiter. Der betreffende Teilnehmer meldet sich auf dem RACH und bekommt über den AGCH (Immediate Assignment Nachricht) einen eigenen SDCCH zugewiesen, auf dem sich MS und MSC über den Aufbau einer Sprachverbindung verständigen. An dieser Stelle erfolgt die im nächsten Kapitel beschriebene Verschlüsselung bzw. Authentifizierung. Sobald dies geschehen ist, schickt das MSC eine Assignement Request an den BSC mit der Anweisung einen TCH aufzubauen. Der BSC aktiviert daraufhin einen freien TCH in der BTS und informiert die MS über den SDCCH über den zugewiesenen TCH. Die MS wechselt dann auf den angegebenen TCH und bestätigt die erfolgreiche Aufnahme über den FACCH an die BTS, welche ebenfalls über den FACCH bestätigt. Abschließend sendet die MS ein Assignement Complete über die BTS (FACCH) an die BSC. [Sau08, vlg. S.43f]

2.3.2 Handover

Bewegt sich ein Teilnehmer aus der Reichweite einer BTS hinaus, muss er an eine Funkzelle mit besserer Signalqualität weitergereicht werden. Diese Form der Weiterreichung nennt man Handover. Ein Handover wird von dem BSC anhand der über den SACCH gemessenen Signalwerte der aktuellen Zelle und der Nachbarzellen veranlasst. Aufgrund dieser Messdaten kann der BSC entscheiden, in welche Zelle die MS wechseln soll. Bevor der Wechsel erfolgen kann muss der BSC in der neuen BTS eine TCH aktivieren. Erst danach schickt die BSC über die alte BTS das Handoverkommando via FACCH. Dieses Kommando enthält die neue Frequenz und die Timeslot-Nummer des neuen TCHs. Daraufhin kann sich die MS mit der neuen BTS synchronisieren, indem sie in vier aufeinanderfolgende Burst eine Handover Access Nachricht sendet. Im fünften Burst sendet der Teilnehmer eine SABM Nachricht (gesicherte Verbindungsaufforderung) an die BTS, welche bei korrekter Erkennung eine Bestätigung an den BSC schickt. Der BSC muss abschließend noch den alten TCH abbauen. Wenn sich die neue Zelle außerhalb des Einflussbereichs des BSC befunden hätte, müsste für ein erfolgreiches Handover das zuständige MSC mit einbezogen werden, da die BSCs nicht untereinander verbunden sind. Bei einem Handover während einer aktiven Verbindung (wie hier beschrieben) wird abschließend noch die MSC informell benachrichtigt, selbst wenn das Handover innerhalb eines BSC stattgefunden hat. [Sau08, vgl. S.45f, 68f]

3 Verschlüsselung

GSM verwendet zur Verschlüsselung der Nutzdaten den A5/1 Algorithmus. Die Ver- und Entschlüsselung der GSM Nutzdaten findet in der BTS und im Mobiltelefon statt. Des Weiteren fordert das GSM-Netz eine Authentifizierung der Teilnehmer, wenn ein Gespräch aufgebaut werden soll oder ein Location Update stattgefunden hat. Dieses Kapitel erläutert den Verschlüsselungsmechanismus und die Authentifikation innerhalb des GSM-Netzes. Darüber hinaus werden Angriffsmöglichenkeiten auf den verbreitesten GSM-Kryptoalgorithmus A5/1 theoretisch vorgestellt.

3.1 Authentifikation

Die Authentifikation innerhalb eines GSM-Netzes steht nicht im Mittelpunkt dieser Arbeit, wird aber aus Gründen der Vollständigkeit kurz beschrieben. Potentielle Sicherheitslücken wie z.B. das Auslesen des geheimen Schlüssels und das damit verbundene Klonen von SIM Karten[1] werden hier nicht näher betrachtet. Wie bereits im vorangegangenen Kapitel angesprochen verfügt jeder Teilnehmer über einen geheimen Schlüssel k_i, welcher nur auf der SIM Karte und im HLR bzw. Authentication Center (AC) gespeichert ist. Dieser 128 Bit lange Schlüssel wird niemals übertragen, sondern dient als Grundlage für den Ciphering Key k_c und ist mit dem private-Key von PGP[2] vergleichbar. Ein Teilnehmer authentifiziert sich dabei wie folgt: Bei einer Verbindungsaufnahme fordert das MSC anhand der IMSI des Teilnehmers drei Parameter beim HLR bzw. AC an. Eine 128 Bit große Zufallszahl (RAND), die 32 Bit lange Signed Response (SRES) und den k_c für die ggf. später benötigte Verschlüsselung. Die SRES wird im HLR bzw. AC aus der RAND und dem k_i mit Hilfe des A3 Algorithmus berechnet, wobei $SRES = A3(k_i, RAND)$ (siehe Abbildung 3.1, SIM Karte ist blau makiert). Die Funktion, welche A3 zur Berechnung verwendet, muss rechtseindeutig sein, damit bei gleichen RAND und k_i immer die gleiche SRES generiert wird. Der A3 Algorithmus ist nicht standatisiert, sondern Providerab-

[1] http://dasalte.ccc.de/gsm/
[2] Pretty Good Privacy

hängig.[3] Das MSC schickt die RAND an die MS. Diese muss anschließend mit der RAND und ihrem auf der SIM-Karte befindlichen k_i ebenfalls den SRES berechnen. Der A3 Algorithmus wird dabei ausschließlich auf der SIM-Karte ausgeführt, da der k_i die SIM-Karte niemals verlassen darf. Die fertig berechnete SRES schickt die MS anschließend zurück an das MSC, welche nun prüfen muss, ob der SRES vom HLR gleich der SRES der MS ist. Stimmen die beiden SRES überein, ist der Teilnehmer erfolgreich authentifiziert und darf das Netzwerk verwenden. Einem potentiellen Angreifer ist es hier nicht möglich den SRES zu berechnen, da er den k_i nicht kennt. Die einzige Sicherheitslücke der Luftschnittstelle bestünde in der wiederholten Sendung der RAND, welche jedoch bei jedem Authentifizierungsvorgang neu generiert wird und somit ein Aufzeichnen des RAND und SRES nutzlos erscheinen lässt. [Sau08, S.24f]

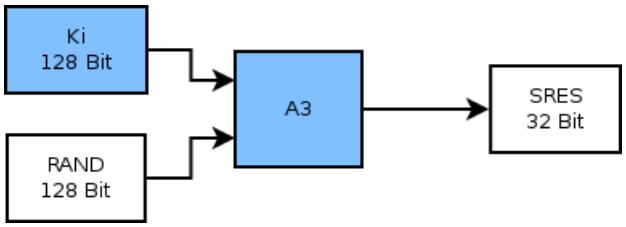

Abbildung 3.1: Erzeugung des SRES nach [Sau08, S.24]

3.2 Verschlüsselungsverfahren

Zur Schlüsselgenerierung kommt der A8 Algorithmus zum Einsatz, welcher anhand der RAND und des k_i den Cipher Key k_c berechnet, sodass $k_c = A8(k_i, RAND)$. Der Ablauf der Schlüsselerzeugung erfolgt im Rahmen der Authentifizierung. Anhand des k_c und der Framenummer berechnet A5/x den Schlüsselframe, welcher mit den Originaldaten *xor*-Verknüpft wird, um den verschlüsselten Frame zu erhalten (Abbildung 3.2). [Sau08, S.58f]

Der Kommunikationsablauf bzgl. Authentifikation und Verschlüsselung ist in Form eines Sequenzdiagramms visualisiert (siehe Abbildung 3.3). Dabei ist zu beachten, dass das AuC, wie im vorherigen Kapitel beschrieben, meistens Teil des HLRs ist.

[3] http://www.gsm-security.net/faq/gsm-a3-a8-comp128-broken-security.shtml

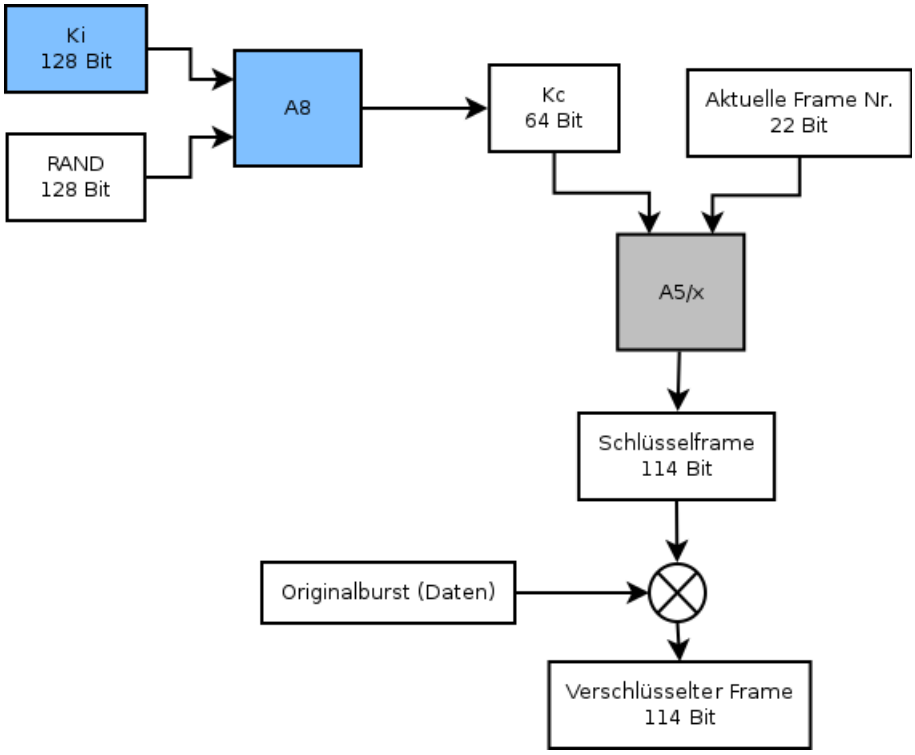

Abbildung 3.2: Erzeugung des Cipher Keys k_c und Schlüsselframes nach [Sau08, S.58]

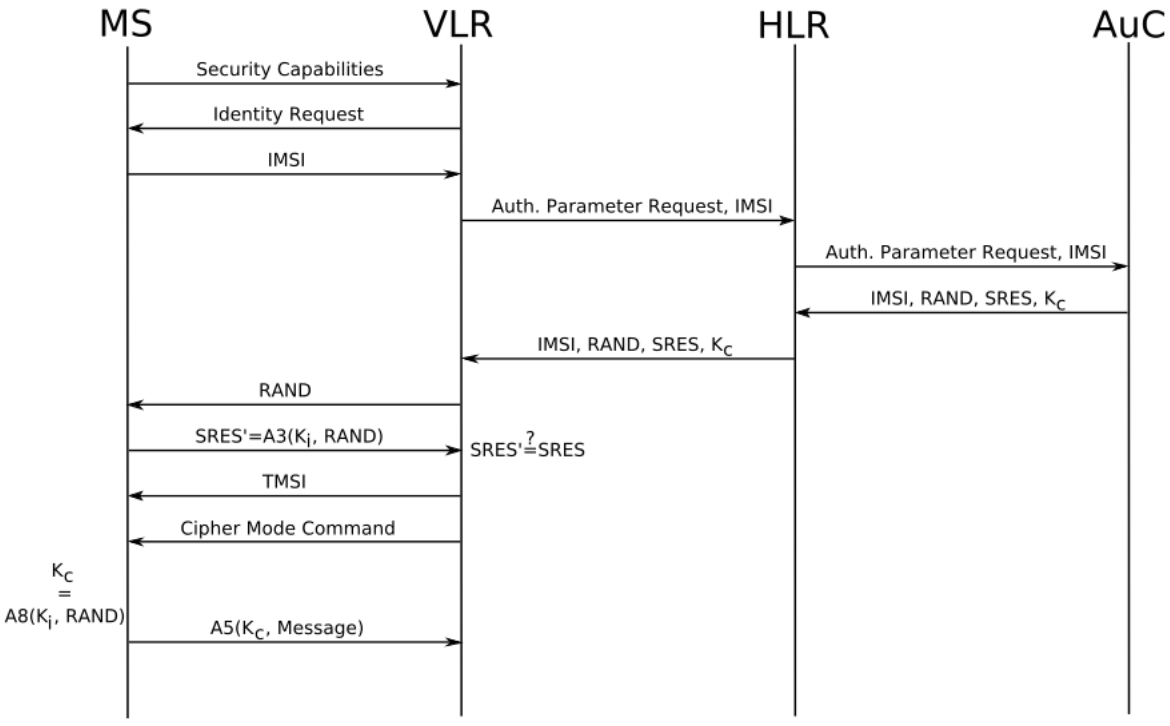

Abbildung 3.3: Ablauf von Authentifikation und Verschlüsselung aus [Str07, S.10]

3.2.1 A5/x Familie

Um die Verschlüsselung möglichst flexibel zu gestalten, spezifiziert GSM mehrere Kryptoalgorithmen. Ein weiterer Grund für multiple Verschlüsselungsalgorithmen besteht darin, dass einige Algorithmen nicht an bestimmte Staaten exportiert werden dürfen. GSM ermöglicht auch das Einführen neuer Algorithmen, wobei zu bedenken ist, dass diese auch in den Endgeräten implementiert sein müssen, was eine Umstellung erschweren würde. Im Allgemeinen ist die Verschlüsselung in GSM optional, was aktive Angriffe wie unten beschrieben zur Folge haben kann. GSM unterscheidet zwischen vier verschiedenen Algorithmen zur Verschlüsselung, von denen vorwiegend nur A5/1 und A5/2 verwendet werden. Der im nächsten Kapitel beschriebene Angriff beschränkt sich auf den A5/1 Algorithmus, weshalb dieser im Folgenden näher betrachtet wird.

A5/0 A5/0 ist lediglich eine Bezeichnung für das Unterlassen der Verschlüsselung. Ist eine BTS keiner Verschlüsselungstechnik mächtig oder gibt dies lediglich vor, findet die Kommunikation unverschlüsselt statt.

A5/1 A5/1 ist der in den meisten Staaten am häufigsten verwendete Algorithmus, welcher nie offiziell publiziert wurde. 1999 veröffentlichten M. Bricenco, I. Goldberg und D.Wagner eine genaue Beschreibung der Chiffre, die trotz kleiner Unsicherheiten bezüglich der Schlüsselframegröße allgemein anerkannt ist. [Sü03, S.11]

Die Stromchiffre basiert auf Modulo-2 Addition und besteht aus drei Schieberegistern (LFSR[4], siehe Abbildung 3.4). Die Schieberegister sind 19, 22 und 23 Bits lang und werden an den Ausgängen mittels XOR zusammengeführt. Eine dynamische Taktsteuerung wählt zufällig die zu taktenden Register. Vor Beginn der Verschlüsselung werden die drei Register mit dem Nullvektor initialisiert. Im Anschluss wird der Sitzungsschlüssel k_c bitweise in die Register kopiert. Da k_c 64 Bit lang ist, wird jedes Register 64 mal getaktet, sodass mit jedem Takt ein Bit von k_c kopiert wird. Da die Registergrößen kleiner als 64 Bit sind, werden die überschüssigen Bits auf den Registern links „herausgeschoben“ und mittels XOR wieder auf der rechten Seite „reingeschoben“. Danach wird mit dem gleichen Verfahren die aktuelle Framenummer in jedes Register geladen, womit A5/1 initialisiert ist und mit der Verschlüsselung begonnen werden kann.

Um der Chiffre ihre Linearität zu nehmen[5], wird nun von der Taktsteuerung gebrauch gemacht. Die LFSRs können zu verschiedenen Zeitpunkten angesprochen

[4] Linear Feedback Shift Register

[5] Ansonsten wären die Ausgänge der LFSR verknüft, sodass durch das Lösen eines linearen Gleichungssystems der Initialzustand rekonstruiert werden könnte.

werden. Die drei sog. Taktkontrollbits befinden sich (laut Annahme) bei dem ersten LFSR bei Register 11, beim zweiten LFSR bei 12 und beim dritten bei Register 13. Dabei wird wie folgt getaktet: Hat höchstens ein Taktkontrollbit den Wert 1, takten alle LFSRs deren Taktkontrollbits auf 0 stehen. Ist mehr als ein Taktkontrollbit auf 1, takten alle LFSR, deren Taktkontrollbits auf 1 stehen. Wie der Abbildung 3.4 zu entnehmen ist, werden für die Verschlüsselung nicht alle 328 Bit genutzt. Die ersten 100 Bit verfallen, während die nachfolgenden 114 Bit auf den Datenstrom aufaddiert werden. Die Restlichen 114 Bits werden xor verknüft. [Sü03, S.11ff]

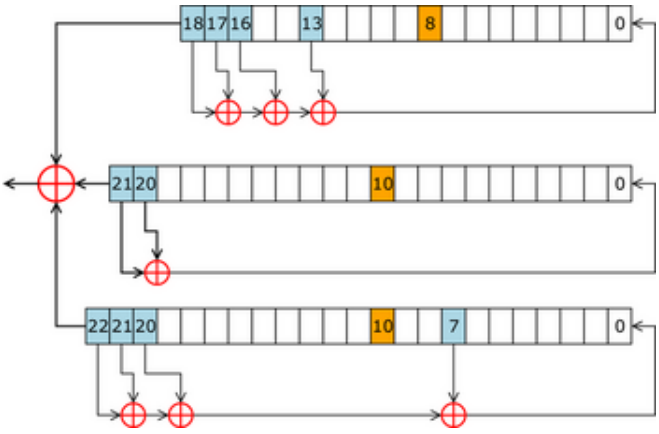

Abbildung 3.4: Funktion der A5/1 Chiffre[6]

A5/2 Der A5/2 Algorithmus ist eine abgeschwächte Variante[7] von A5/1 und ist für die Staaten mit Exportverbot für Sicherheitstechniken vorgesehen. Die Verwendung dieses Algorithmus ist seit 2007 von der 3GPP nicht mehr empfohlen, da eine Brechung des Schlüssels innerhalb von Sekunden möglich ist. [Ger12, Folie 52]

A5/3 Bei A5/3 handelt es sich um eine 2002 hinzugefügte Blockchiffre, die auf dem offenen Kasumi Algorithmus basiert, der von der 3GPP [8] definiert wurde. Dieser Algorithmus bildet die Kernverschlüsselung für UMTS. Die Schlüssellänge ist von 64 bis 128 Bits standartisiert. [3GP00, S.6] und gilt laut dem Kryptologen David Wagner[9] als sehr sicher. [Qui04, S.19]

A5/4 Bei A5/4 handelt es sich ebenfalls um einen neueren Standard, welcher jedoch auf kaum einer Basisstation implementiert ist. An dieser Stelle sei angemerkt, dass die

[6] Abbildung aus: http://edipermadi.files.wordpress.com/2008/03/a5_cipher.png
[7] Es existiert ein viertes LFSR für die Taktkontrollbits
[8] 3rd Generation Partnership Project, Zusammenschluss von Standardisierungsgremien im Mobilfunkbereich
[9] https://groups.google.com/forum/?fromgroups=#!topic/sci.crypt/n5UGac7Ky6k

Verschlüsselung in den BTS in Hardware (z.B. FPGAs) umgesetzt ist und somit eine Änderung oder ein Update des Verschlüsselungsalgorithmus hohe Kosten für die Betreiber bedeuten würden. Verwendet wird der gleiche Algorithmus wie bei A5/3 mit einer Schlüssellänge von 128 Bit. [3rd09, S.9]

3.3 Angriffsvektoren

Angriffe können in aktive und passive Angriffe unterschieden werden. Sie definieren lediglich den Interaktionsmodus des Angreifers, wobei dieser im aktiven Modus mit dem anzugreifenden System interagiert, im passiven Modus jedoch nicht. Ein passiver Angriff hat somit den Vorteil, nicht entdeckt zu werden.

3.3.1 Aktive Angriffe

Da aktive Angriffe im Allgemeinen nicht unentdeckt bleiben, werden im Folgenden nur mögliche Szenarien beschrieben, jedoch nicht weiter ausgeführt. Eine Möglichkeit, ein GSM Netz anzugreifen, besteht in der Emulation einer BTS. Das Netz ist aus Sicht des Teilnehmers stets vertrauenswürdig und muss sich im Gegensatz zum Teilnehmer nicht authentifizieren. Wenn eine BTS gegenüber dem Teilnehmer angibt, keine der möglichen Verschlüsselungen zu beherrschen, so wird dies von der Mobilstation akzeptiert und die Kommunikation findet im Klartext statt. Ein potentieller Angreifer kann sich diesem Umstand zu Nutze machen, indem er mit entsprechende Hardware[10] und Software[11] eine solche BTS emuliert, die alle Teilnehmer zur Klartextkommunikation zwingt, an das echte Netz weiterleitet und dabei für den Angreifer relevante Informationen extrahiert.

Eine weitere Möglichkeit GSM zu attackieren basiert auf einem Denial of Service (DoS) Angriff. Dabei stellt der Teilnehmer in sehr kurzen Intervallen über den RACH (Random Access Channel) eine Gesprächsanfrage, ignoriert jedoch die Immediate Assignement Bestätigung. Die BTS ist schnell mit den Anfrage überfordert, sodass sich neue Teilnehmer an einer Nachbarstation anmelden müssen. Dieser Angriff kann zum einen dafür genutzt werden Mobilstationen auf die eigene, möglicherweise nicht so sendestarke BTS zu lenken. Zum anderen können damit auch flächendeckend GSM Netze außer Gefecht gesetzt werden, sofern man diesen Angriff für jede BTS eines Areals durchführt. [Ger12, Folie 60]

[10] http://www.fh-kl.de/ andreas.steil/Projekte/OpenBTS/index.html#Hardware
[11] http://gnuradio.org/redmine/projects/gnuradio/wiki/OpenBTSBackground

3.3.2 Passive Angriffe

Passive Angriffe bleiben meist unentdeckt, da sie per Definition keinen Einfluss auf das anzugreifende System nehmen. Es gab in der Vergangenheit mehrere Angriffe auf A5/1, von denen einige Ideen hier kurz beschrieben werden. Eine dieser Ideen bildet die Grundlage des im praktischen Kapitel dargestellten Angriffs.

- Eine der vergleichsweise einfacheren Angriffe veröffentlichten die Kryptologen M. Briceno, I.Goldberg und D. Wagner. Sie stellten fest, dass die Länge des Sitzungsschlüssels k_c meist von 64 Bit auf 54 Bit gekürzt wird. Somit könnte man mit Hilfe eines Brute-Force-Angriffs den richtigen Schlüssel nach 2^{53} Versuchen erraten. Der Aufwand beträgt also $O(2^{54})$. Dieser Angriff wird auch als *Vollständige Suche* bezeichnet. [Sü03, S.14]

- J. Golic verfolgte die Idee A5/1 auf der Basis vorberechneter Datensätze anzugreifen. Dabei bediente er sich dem Phänomen des Geburtstagsparadoxons. Das Geburtstagsparadoxon beschäftigt sich mit der Frage, wie groß eine Gruppe von Personen sein muss, damit die Wahrscheinlichkeit, dass zwei Personen am selben Tag Geburtstag haben größer als 50% ist. 23 Personen sind aus stochastischer Sicht dafür ausreichend.[12] Golic dreht diese Fragestellung um und fragt: „Wieviele Personen müssen im Durchschnitt befragt werden, damit die Chance größer 50% ist, daß der Fragende und der Befragte den selben Geburtstag haben?“ [Sü03, Zitat S.15] Bezogen auf A5/1 fragt sich Golic, wie viele Frames von maximal 2^{64} er in einer Datenbank speichern müsste, damit die Wahrscheinlichkeit einer Kollision mit potentiell abgefangenen Frames bei fast 100% liegt. Je länger man ein Gespräch aufzeichnet, desto höher ist die Wahrscheinlichkeit einer Kollision. Dieses Verfahren bezeichnet man als Time-Memory-Tradeoff, da es vorberechnete Daten (Memory) und den zeitlichen Aufwand (Time) in ein antiproportionales Verhältnis setzt. Die Angriffsvariante erwies sich als ineffizient, da selbst bei großen Vorberechnungen eine lange Gesprächsdauer von Nöten ist. [Sü03, S.15]

- Die Idee von Golic wurde von A. Biryukov, A. Shamir und D. Wagner aufgegriffen, jedoch unter Berücksichtigung bestimmter Kriterien, wie z.B. die regelmäßige Neuinitialisierung mit dem selben Schlüssel. Mit diesen Modifikationen konnte der zeitliche Aufwand deutlich reduziert werden. Sogar Echtzeitbrechungen wären mit dieser Variante mit sehr hoher Rechenleistung möglich. [Sü03, S.15f]

[12] Mathematischer Hintergrund:
http://www.members.dokom.net/holger.schroeder/Literaturverzeichnis/Didaktik%20der%20Stochastik.pdf

Aus dem Time-Memory-Tradeoff-Angriff nach J. Golic und den Modifikationen von Biryukov, Shamir und Wagner enstand die Idee der Rainbowtable, welche unter anderem von dem deutschen Kryptospezialisten Karsten Nohl [13] im Rahmen eines Open Source Projektes[14] berechnet und verbreitet wurde. Sowohl die Rainbowtables als auch Programme zu deren Kreation bzw. Berechnung wurden im Rahmen des Projekts veröffentlicht.

3.3.3 Rainbowtable

Eine Rainbowtable ist eine vorberechnete Datenstruktur, welche eine effiziente probabilistische Suche nach dem Klartext zu einem Hashwert ermöglicht. Dabei wird kein herkömmliches Brute-force Verfahren, wie beim Time-Memory-Tradeoff durchgeführt, sondern eine verbesserte Variante. Die Rainbowtable besteht aus mehreren Ketten, welche mit einem Klartextwort starten und einem Hashwert enden. Innerhalb einer Kette wird das Startwort gehashed und der entstandene Hash wiederum zu einem Klartextwort reduziert. Dieses Klartextwort wird wieder gehashed, dann wieder reduziert etc., bis zu einer festgelegten Iteration (siehe Abbildung 3.5). Dabei werden innerhalb einer Kette unterschiedliche Reduktionsfunktionen verwendet. Bei Berechnung der Ketten darf kein Wort, welches innerhalb der Kette vorkommt als Startwort verwendet werden, da es sonst zu Kollisionen kommt. [Sü06] [Oec03]

Nach der Berechnung der Ketten wird lediglich das Startwort und der letzte Hash ge-

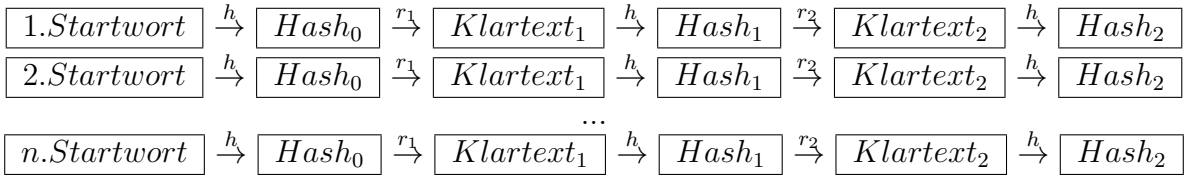

Abbildung 3.5: Aufbau einer Rainbowtable

speichert. Dies spart Speicherplatz, auf Kosten von vergleichsweise geringer Rechenzeit. Grundsätzlich besteht die Rainbowtable aus mehreren berechneten Tabellen, die in ihr als Spalten vorkommen. Die Berechnung mehrerer Tabellen ist notwendig, um Kollisionen zu vermeiden, welche die Rechenzeit bei der späteren Schlüsselsuche stark erhöhen würde. Die verschiedenen Tabellen (jede Tabelle könnte eine Farbe des Regenbogens repräsentieren) unterscheiden sich in der Reduktionsfunktion. Daher ergibt sich in der spaltenweisen Anordnung dieser Tabellen eine Rainbowtable (jede Spalte stellt eine Far-

[13] http://www.heise.de/security/meldung/GPRS-Verbindungen-leicht-abhoerbar-1320879.html
[14] http://www.reflextor.org/trac/a51

be/Tabelle dar), die nach jedem Hash innerhalb einer Kette eine andere Reduktionsfunktion aufweist. Bezogen auf A5/1 repräsentiert die Hashfunktion die Verschlüsselung mit A5/1, sodass gegebende verschlüsselte Daten in der Rainbowtable gesucht werden. Dabei wendet man auf den verschlüsselten Frame so lange die Reduktions- und Hash- bzw. Verschlüsselungsfunktionen an, bis ein Wert gefunden wurde, der in der letzten Spalte der Rainbowtable zu finden ist. Damit ist die Kette gefunden, in welcher nach dem Klartext zu dem verschlüsselten Frame zu suchen ist. Schließlich wendet man Hash- und Reduktionsfunktionen auf den Startwert der gefundenen Kette an, bis der Hashwert und damit der zugehörige Klartext erreicht wurde. [Mey10]
Im Folgenden sei ein kleines Beispiel beschrieben, welches das Finden des Klartextes zu einem verschlüsselten Wort anhand eines Auszugs aus einer Rainbowtable (siehe Abbildung 3.6) zeigt. Sei *re3xes* das Wort, dessen Klartext anhand der Rainbowtable ermittelt werden soll. Im ersten Schritt werden die Funktionen (Hash und Reduktion) einer Kette angewand, um ein Klartextwort zu finden, welches sich auf der rechten Seite der Rainbowtable befindet. Ein Auszug der Rainbowtable ist auf der linken Seite der Abbildung 3.6 dargestellt. Die Suche in der ersten Kette ist jedoch erfolglos und wird daher in einer weiteren Kette fortgesetzt (Schritt 2). In der zweiten Kette führt das Finden der Klartextes *linux23* zum Erfolg, da es in der letzen Zeile der Tabelle auf der rechten Seite zu finden ist. Anschließend (Schritt 4) werden die Funktionen für diese Kette, beginnend beim Startwort bis zum verschlüsselten Wort, durchgeführt. Ist dieses gefunden (Schritt 5), kann auch der Klartext durch einfaches Ablesen ermittelt werden. Da das Klartextwort *culture* offenbar zu dem gesuchten Schlüsselwort *re3xes* verschlüsselt wird, ist der gesuchte Klartext gefunden.

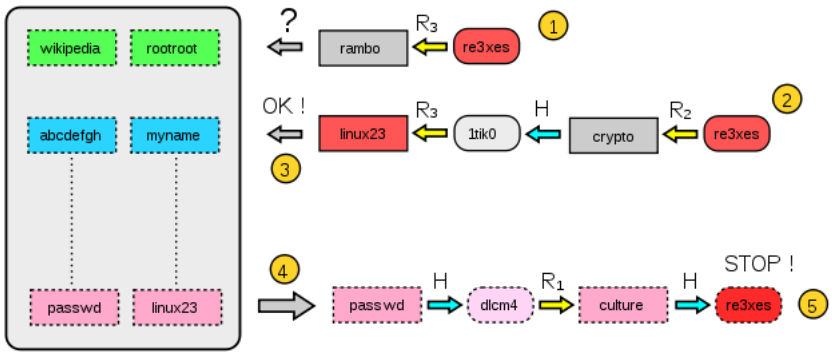

Abbildung 3.6: Beispiel zur Anwendung einer Rainbowtable[15]

GSM vereinfacht zusätzlich einen Angriff auf A5/1, da es Nachrichten, welche einen

[15] Abbildung aus: http://upload.wikimedia.org/wikipedia/commons/thumb/9/93/Rainbow_table2.svg/650px-Rainbow_table2.svg.png

Frame nicht vollständig ausfüllen mit bekannten Füllzeichen (Paddingbits, z.B. 2b..2b) ergänzt, die sowohl verschlüsselt als auch im Klartext verschickt werden. Dieser Umstand erleichert die Berechnung des Sitzungsschlüssels k_c deutlich. Selbiges geschieht mit Broadcastnachrichten, welche zu bekannter Zeit verschickt werden und somit die Vertraulichkeit der gesamten Kommunikation ausdrücklich gefährden. Unter anderem kann bei folgenden bekannten Nachrichten die Position im verschlüsselten Bereich ermitelt werden [Bet11, Folie 13]:

- System Information Nachrichten
- $LAPD_m$ Nachrichten
- Acknowledge Nachrichten
- Call Proceeding, Alerting, Connect Nachrichten

4 Praktische Angriffsmöglichkeiten

Nachdem die theoretischen Hintergründe der Sicherheitsproblematik der GSM Luftschnittstelle erläutert wurden, stellt sich die Frage ob, und mit welchen Mitteln eine praktische Ausnutzung der Sicherheitslücken denkbar ist. Ziel dieses Kapitels ist es aufzuzeigen, welche Schritte erforderlich sind, um mit geringem Kostenaufwand die Vertraulichkeit der über GSM geführten Kommunikation zu brechen.

4.1 OsmocomBB-Projekt

OsmocomBB steht für *Open source mobile communications BaseBand* und ist eine Open Source Implementierung der Firmware für den GSM Baseband Prozessor. Es implementiert Treiber für das GSM Baseband sowie den clientseitigen GSM-Protokollstack von Schicht 1 bis 3 [Wel12] Ziel des Projektes ist es, über eine offene Implementierung der Grundfunktionalitäten eines Mobiltelefons zu verfügen. Dabei spielt die Kontrolle über das eigene Mobiltelefon eine zentrale Rolle. Insbesondere in Anbetracht der Tatsache, dass ein Provider Updates auf die eigene SIM-Karte aufspielen und ausführen kann, ohne die Kenntnis des Mobiltelefonbesitzers. Dieses Vorgehen basiert ausschließlich auf dem Vertrauen zu dem Provider. Eine ähnliche Problematik besteht in dem Senden von sogenannten *Silent-SMS*. Dabei handelt es sich um eine SMS, welche still empfangen wird, also das Mobiltelefon dem Anwender den Emfang der SMS verschweigt. Dieser Umstand kann beispielsweise bei einer Ortung des Mobiltelefons mittels einer Silent-SMS[1] für die ermittelnde Behörde von Vorteil, für die geortete Person ein Nachteil sein.

4.2 Notwendige Hard- und Software

Im folgenden sollen die Hard- und Softwarekomponenten genannt und erläutert werden, die für einen Angriff auf die GSM Luftschnittstelle notwendig sind. Vorrausgesetzt wird ein Laptop mit einem Linux-Betriebsystem.

[1] Der Abstandsradius der MS von der BTS kann damit ermittelt bzw. eingeschränkt werden

4.2.1 Vorbereitungen zum Aufzeichnen von GSM-Bursts

Zunächst benötigt man einen GSM-Empfänger, welcher in der Lage ist GSM-Nutzdaten aufzuzeichnen, die nicht für ihn bestimmt sind. Die Hardware eines gewöhnlichen Mobiltelefons eignet sich als GSM-Empfänger, hat jedoch die Eigenschaft, nur die eigenen Verbindungen zu verfolgen. An dieser Stelle hilft das oben beschriebene OsmocomBB-Projekt, das es erlaubt, die GSM-Kommunikation bestimmter Mobiltelefone [2] zu kontrollieren. Für die folgenden praktischen Versuche wird ein Motorola C123 verwendet. Die Kommunikation erfolgt über eine serielle Schnittstelle, welche sich an der Audioschnittstelle (Headset, siehe Abbildung 4.1) befindet. Die physikalische Verbindung zu

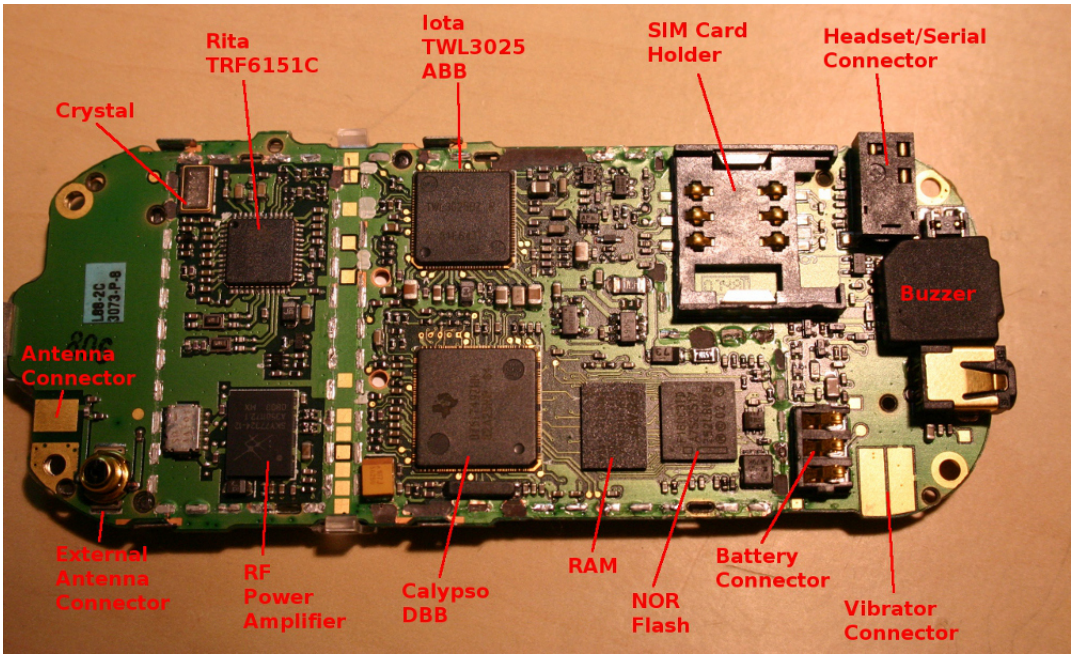

Abbildung 4.1: Motorola C123 Board[3]

einem Laptop wird mittels eines USB↔Seriell-Adapters hergestellt. Die von OsmocomBB bereitgestellte Firmware muss zuvor kompiliert[4] werden, ggf. ist vorher eine Anpassung der Baudrate in der Firmware notwendig. Anschließend kann über das Tool *osmocon*[5] die Firmware in den Arbeitsspeicher des Mobiltelefons geladen und gestartet werden. Osmocon abstrahiert den GSM Basebandprozessor über die Firmware und stellt somit eine Schnittstelle für Applikationen (unix socket, siehe Abbildung 4.2) oberhalb der ersten Schicht dar. Die Terminalausgabe von osmocon zeigt die wichtigsten Aktivitäten auf der

[2] Unterstützt werden Mobiltelefone u.a. die Modelle C118, C120, C121, C123 und C155 von Motorola
[3] http://bb.osmocom.org/trac/attachment/wiki/MotorolaC123/c123_pcb.jpg
[4] Für i368 oder amd64 Architekturen muss ein ARM cross compiler verwendet werden.
[5] http://bb.osmocom.org/trac/wiki/osmocon

untersten GSM-Schicht. Für den praktischen Angriff wird eine angepasste Version[7] von

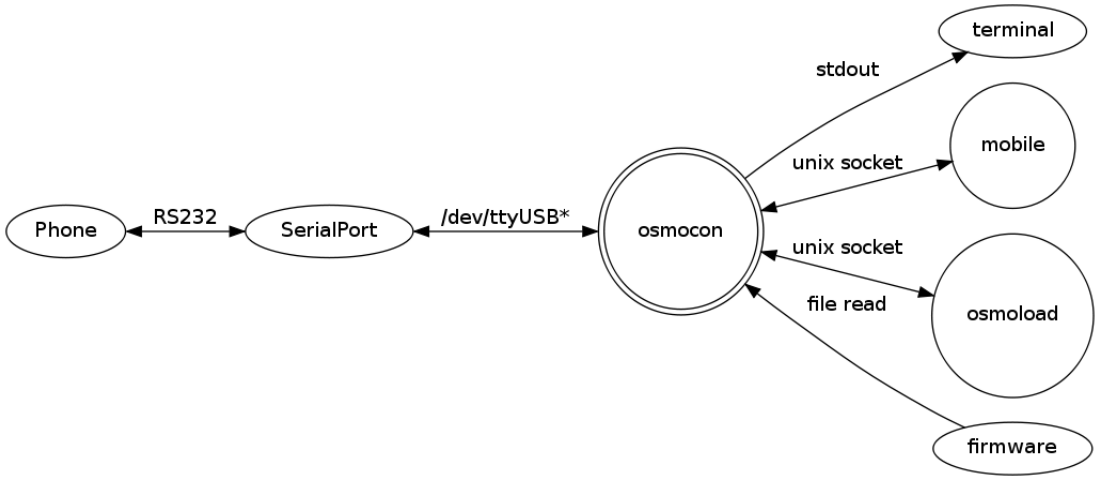

Abbildung 4.2: Interaktion mit osmocon[6]

OsmocomBB verwendet, die auf das Mitschneiden verschlüsselter GSM-Bursts optimiert ist. Die Vorbereitungen zum Mitscheiden von GSM-Bursts sind getroffen. Bevor der Angriff gestartet wird, erfolgt die Vorbereitung für die Entschlüsselungsphase.

4.2.2 Vorbereitungen zum Entschlüsselung der GSM-Bursts

Zunächst benötigt man die bereits oben erläuterte Rainbowtable für A5/1. Die Algorithmen zum Generieren dieser Hashtabelle sind öffentliche verfügbar [8], jedoch dauert die Berechnung auf hochwertigen Grafikkarten zur Zeit mehrere Monate, sofern man die Rechenlast auf mehrere Computer verteilt. [9] Karsten Nohl veröffentlichte 2010 eine vorgenerierte Rainbowtable, die u.a. über das Torrentnetzwerk erreichbar ist. [10] Diese Hashtabelle wird im Folgenden verwendet, um GSM-Bursts zu entschlüsseln. Zu diesem Zweck gibt es das Tool *Kraken*[11], welches A5/1-Schlüsselframes in der Rainbowtable sucht und anhand der gefundenen Klartexte den Sessionkey k_c berechnet. Kraken stellt u.a. auch die Algorithmen zur Berechnung der Rainbowtable auf der CPU oder ATI-Grafikkarten bereit. Bevor Kraken die Tabelle zum Entschlüsseln benutzen kann, muss diese zuvor konvertiert werden. Im Rahmen der Konvertierung schreibt sich Kraken für jede der 40 Tabellen (.dlt-Endung) einen Index (ca. 80 MB pro Datei) in das eigene Verzeichnis. Die

[6] http://bb.osmocom.org/trac/graphviz/64ec4528d90377aa7cc41e9eb6b3c57a4da13675.dot.png
[7] Erreichbar unter https://github.com/DrWhax/osmocom-bb-raw
[8] http://www.reflextor.org/trac/a51/browser/tinkering
[9] http://www.rechenkraft.net/wiki/index.php?title=A51_(beendet)
[10] http://opensource.srlabs.de/projects/a51-decrypt/files
[11] http://www.reflextor.org/trac/a51/browser/tinkering

gesamte Tabelle wird direkt [12] auf eine (oder mehrere) Festplatte(n) geschrieben, wobei die Position der Tabellen anhand eines Offsets in einer Konfigurationsdatei (tables.conf) gespeichert wird. Zuvor wurde(n) die Zielfestplatte(n) in der tables.conf eingetragen, welches das folgende Kraken-Skript benutzt, um den beschriebenen Konvertierungsvorgang durchzuführen:

```
./Behemoth.py <Verzeichnis der Rainbowtable (.dlt-Endung)>
```

Sämtliche Vorbereitungen zum Mitschneiden und Entschlüsseln von GSM-Daten sind getroffen, sodass mit der Angriffsphase begonnen werden kann.

4.3 Angriffs-Szenario

Das Angriffszenario kann in die Aufzeichnungsphase (Sniffing) und den Entschlüsselungsprozess unterteilt werden.

4.3.1 Aufzeichnen von GSM-Bursts (Sniffing)

Ist osmocon gestartet[13] und Layer 1 auf dem Mobiltelefon aktiviert, kann mit der Arbeit auf den höheren Schichten begonnen werden. Dabei soll zunächst die Frequenz bzw. ARFCN eines aktiven Kanals gefunden werden. Zu diesem Zweck kann die Osmocom-BB Applikation *cell_log* genuzt werden.

```
./cell_log
```

Die Applikation *cell_log* sucht automatisch aktive Frequenzen und synchronisiert sich mit signalstarken Zellen, um detailliertere Informationen bzgl. MCC und MNC (Provider) zu erhalten. Ein Ausschnitt der Messungen könnte beispielsweise wie folgt aussehen:

```
<000e> cell_log.c:367 Measure from 0 to 124
<000e> cell_log.c:367 Measure from 512 to 885
<000e> cell_log.c:367 Measure from 955 to 1023
<000e> cell_log.c:358 Measurement done
<000e> cell_log.c:340 Sync ARFCN 102 (rxlev -53, 570 syncs left)
<000e> cell_log.c:190 Cell: ARFCN=102 MCC=262 MNC=01 (Germany, T-Mobile)
```

[12] direkt bedeutet, dass die Daten nicht in ein eingehangendes Verzeichnis einer Festplatte, sondern direkt auf ein Festplattendevice, z.B. /dev/sdb1 ohne Dateisystem geschrieben werden

[13] Genaue Instruktionen sind dem ausführlich dokumentierten Projekt zu entnehmen: http://bb.osmocom.org/trac/wiki/osmocon

```
<000e> cell_log.c:340 Sync ARFCN 12 (rxlev -65, 569 syncs left)
<000e> cell_log.c:190 Cell: ARFCN=12 MCC=262 MNC=02 (Germany, Vodafone)
```

Aus der Programmausgabe kann die ARFCN aktiver Frequenzen ausgelesen und für den Angriff verwendet werden. An dieser Stelle kann ausgewählt werden, welches Providernetz als Angriffsziel dienen soll. Optional kann mit einer weiteren Applikation von OsmocomBB verifiziert werden, ob auf einem der oben aufgelisteten Kanäle tatsächlich in Bezug auf Nutzdaten aktiv ist.

```
./mobile -i 127.0.0.1
```

Die Mobile Applikation sucht automatisch aktive Frequenzen und synchronisiert sich mit einer signalstarken Zelle. Anschließend kann das Paging-Verhalten auf diesem Kanal (PCH, RACH, AGCH) beobachtet werden (Abbildung 4.3 [14]). An dieser Stelle können *Immediate Assignment* Nachrichten gelesen und folglich auch gefiltert werden. [15] Dabei gibt es zwei Typen von Immediate Assingnments, die im Folgenden in Form eines Beispiels aufgelistet sind:

- `ta 3/1661m ra 0x7f chan_nr 0x0f MAIO 1 HSN 62 TS 7 SS 0 TSC 5`
- `ta 1/553m ra 0x0f chan_nr 0x42 ARFCN 102 TS 2 SS 0 TSC 5`

```
<000b> gsm48_rr.c:2174 PAGING REQUEST 1
<000b> gsm48_rr.c:2132  TMSI ce4394e7 (not for us)
<000b> gsm48_rr.c:2174 PAGING REQUEST 1
<000b> gsm48_rr.c:2132  TMSI 7309e122 (not for us)
<0001> gsm48_rr.c:2450 IMMEDIATE ASSIGNMENT:
<0001> gsm48_rr.c:2462  (ta 2/1107m ra 0x79 chan_nr 0x0e MAIO 1 HSN 62 TS 6 SS 0 TSC 5)
<0001> gsm48_rr.c:2478 Not for us, no request.
<0001> gsm48_rr.c:2450 IMMEDIATE ASSIGNMENT:
<0001> gsm48_rr.c:2462  (ta 2/1107m ra 0x79 chan_nr 0x0e MAIO 1 HSN 62 TS 6 SS 0 TSC 5)
<0001> gsm48_rr.c:2478 Not for us, no request.
<000b> gsm48_rr.c:2174 PAGING REQUEST 1
<000b> gsm48_rr.c:2132  TMSI c341c22b (not for us)
<000b> gsm48_rr.c:2174 PAGING REQUEST 1
<000b> gsm48_rr.c:2141  IMSI 37002[redacted] (not for us)
```

Abbildung 4.3: Ausschnitt der Ausgabe von mobile

Der Parameter *ta* (timing advance) gibt die Entfernung des Teilnehmers von der BTS an. Dabei errechnet sich die Entfernug mit $r = 300m + ta \cdot 550m$, sodass ein ta-Wert von 1 einem Radius von ca. 850 Meter entspricht. [Gö03, S.22] Zu den obigen Meter-Angaben im Immediate Assignment müssen also noch 300m addiert werden. Ein weiterer,

[14] Die IMSI, welche laut MCC (370) aus der Dominikanische Republik stammt, wurde aus Datenschutzgründen unkenntlich gemacht.

[15] Das Filtern der Immediate Assignments ist eine eigene Ergänzung

wichtiger Wert ist die *chan_nr* (Channel-Nummer) des SDCCHs, auf dem die weitere Kommunikation stattfinden soll. Die Begriffe $MAIO$ und HSN wurden bereits weiter oben im Rahmen des Frequencyhoppings erläutert. Die Variablen ra (rate adaption), TS, SS und TSC (training sequence code) sind Kontroll- und Regulationsparameter, die für das weitere Vorgehen nebensächlich sind. Die zweite Immediate Assignment Nachricht enthält anstatt Informationen über ein Frequencyhopping die ARFCN, welche auf die Kommunikationsfrequenz hinweist.
Mit der (hier optional verifizierten) Information über die Frequenz und der dortigen Aktivität wird anschließend ein weiteres Tool des Osmocom-BB Projektes gestartet.

```
./ccch_scan -s /tmp/osmocom_l2 -a 102 -m xml
```

Das Programm *ccch_scan* folgt nun allen Immediate Assignments auf dieser Frequenz (102) in den SDCCH, zeichnet sämtliche Bursts auf, die dieser Kommunikation angehören und speichert diese in einem XML-Format:

```
<scan arfcn="102">
    <frame uplink="0" cipher="1">
        <burst fn="1161098">
            <cyphertext>
                11101110111101101010111010101110010000110010010110...
            </cyphertext>
            ...
        </burst>
        ...
    </frame>
    ...
</scan>
```

Schließlich verfügt man über verschlüsselte GSM-Bursts, welche es im Anschluss zu entschlüsseln gilt.

4.3.2 Entschlüsselung der GSM-Bursts

Zunächst wird die Applikation Kraken gestartet, welche alle vorhandenen Tabellen-Indizes einliest und über einen angebenen Port für andere Programme erreichbar ist:

```
./kraken ../indexes <lokaler port>
```

Nachdem Kraken aktiv ist und auf zu brechende A5/1-verschlüsselte Bitfolgen wartet, kann ein Skript gestartet werden, welches die in XML gespeicherten Bursts einliest und anhand einer Vorhersage versucht den Schlüsselframe zu errechnen.

```
./gsmcrack.py <scan-datei> <prediction>
```

Eine *prediction* ist die Vorhersage, welche Art von Broadcast ein Frame beinhalten könnte. Die Broadcastnachtichen werden abhänging von der BTS in bekannten Zeitschlitzen gesendet, sodass ein Klartext und der dazugehörige verschlüsselte Text vorliegt. Eine Broadcastnachticht des SDCCHs, welche lediglich als aussageleere Füllnachricht verwendet wird, sieht beispielsweise wie folgt aus [16]:

```
0303012b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b
```

Das Wissen darüber, welcher Hersteller die Ziel-BTS gebaut hat und damit welche Füllnachricht gesendet wird, verkürzt den nachfolgend beschriebenen Rechenaufwand erheblich, da kein Durchprobieren von herstellerspezifischen Füllnachrichten nötig ist. Wendet man auf beide Texte ein *xor* an, ergibt sich aus $xor(plaintext, ciphertext)$ der Keystream, in Abbildung 3.2 Schlüsselframe genannt. Kommuniziert wird mit dem Kraken via *telnet* und der beim Start von Kraken festgelegten Portnummer. Dabei wird der Schlüsselframe über telnet mit dem Kommando crack übergeben. Kraken sucht anschließend in der Rainbowtable anhand des Schlüsselframes nach einem möglichen verschlüsselten Sessionkey. War die Suche erfolgreich, erfolgt eine Ausgabe mit einem potentiellen Klartext für den Schlüsselframe, dem Offset (hier 23), der internen Crack-ID (hier 4) und die Teiltabelle der Rainbowtable, in welcher Kraken fündig wurde:

```
Found 299284d7c39a2b18 @ 23  #4  (table:420)
```

Bei dem gefundenen Klartext handelt es sich noch nicht um den Sessionkey k_c, sondern viel mehr um den geheimen Zustand innerhalb der A5/1 Funktion bei der Generierung des Schlüsselframes. Abblidung 3.2 ist zu entnehmen, dass A5/1 zusätzlich die aktuelle Framenummer zur Erzeugung des Schlüsselframes berücksichtigt. Daher muss im Anschluss mit dem gefundenen Klartext und der entsprechenden Framenummer (siehe XML-ausgabe, Attribut *fn*) noch der korrekte Sessionkey k_c gefunden werden. Zu diesem Zweck wird die Kraken-Applikation *find_kc* mit dem gefundenen Klartext, dem Offset, zwei Framenummern der zuvor zum Cracken verwendeten Bursts und dem Schlüsselframe, welcher zum Burst der zweiten Framenummer gehört als Parameter aufgerufen:

[16] Die BTS von Nokiaverwenden diese Bitfolge

```
./find_kc <Klartext> <Offset> <Framenr.1> <Framenr.2> <Schlüsselframe>
```

Schließlich errechnet *find_kc* anhand der gegebenen Parameter potentielle Sessionskeys. Diejenigen Keys, welche die Anforderung erfüllen über 104[17] Bits des 114 Bit langen Keystreams korrekt mit A5/1 zu berechnen, zählen als gültiger Sessionkey k_c. Der Erfolgsfall sieht eine Ausgabe der folgenden Form vor (Ausschnitt):

```
Framecount is 3168370
KC(0): 9b 08 7d f9 25 c7 14 8b  mismatch
KC(1): 9a e2 38 12 84 02 38 f4  mismatch
KC(2): 77 20 a6 43 c1 83 b9 22  *** MATCHED ***
KC(3): 62 96 27 77 63 87 8e 4b  mismatch
KC(4): cc e6 fe 51 cb b3 21 5d  mismatch
```

In diesem Beispiel wurde der gültige Sessionkey k_c *7720a643c183b922* gefunden.

Anhand des Sessionskeys müssen die verschlüsselten Bursts im Anschluss entschlüsselt und decodiert werden. Dazu wird die Applikation *burst_decode* verwendet, die als Parameter die Ziel IP-Adresse zur Ausgabe der entschlüsselten Bursts mit GSMTAP-Pseudoheader [18], die Informaion, ob es sich um einen Uplink- oder Downlinkkanal handelt, die Framenummer des ersten Bursts, den Channel-Typ als Integer, den Sessionkey k_c und schließlich den bzw. die zu dekodierenden Bursts:

```
./burst_decode -i 127.0.0.1 -u <Uplink> --fn <Framenr.> -t <Chan.-Typ>
               -k <kc> -b <Burst1> -b <Burst2> -b <BurstN>
```

Sofern die Füllnachricht zu Beginn des Angriffs richtig geraten wurde, liefert *burst_decode* den gewünschten Klartext in zwei Formen. Einerseits werden die entschlüsselten Bursts als binäre Bitfolge ausgegeben, darüber hinaus werden die entschlüsselten Bursts mit Hilfe des GSMTAP-Pseudoheaders über die lokale Schnittstelle geschickt. Mit Hilfe des Netzwerkanalysetools *Wireshark*[19] können auf dem lokalen Interface die GSM-Daten abgefangen und analysiert werden. Wireshark unterstützt den GSM Protokollstack, darunter auch das Um-Interface (Luftschnittstelle). Jeder Layer 2 bzw. LDAP_m-Frame stellt eine Zeile innerhalb Wireshark dar. Wurde beispielsweise eine SMS gesendet, findet man diese in Wireshark, hier in dem Paket 131 (Abbildung 4.4 [20]).

[17] Dem Quellcode von find_kc entnommen

[18] GSM Frames werden mit Hilfe des Pseudoheaders GSMTAP lokal in UPD/IP Paketen verschickt: http://bb.osmocom.org/trac/wiki/GSMTAP

[19] http://www.wireshark.org/

[20] Beispielaufzeichnung von: http://wiki.wireshark.org/SampleCaptures?action=AttachFile&do=get&target=gsm_sms2.xml

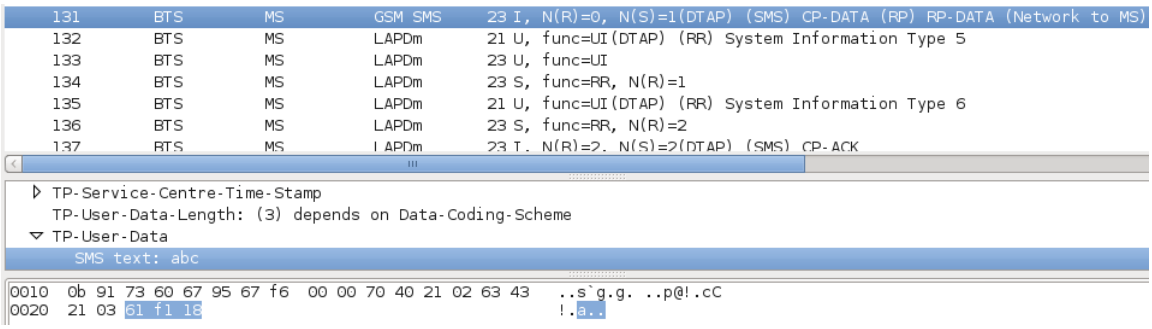

Abbildung 4.4: Eine SMS in Wireshark

Somit wurde eine über GSM versandte SMS erfolgreich abgefangen, entschlüsselt und decodiert. Vorliegendes Beispiel verdeutlicht die Sicherheitsqualität der GSM Luftschnittstelle und stellt klar, dass jede über GSM versandte SMS nicht mehr als vertrauliche Nachricht angesehen werden kann. Selbiges gilt für Sprachverbindungen, wobei der Mehraufwand in dem Frequencyhopping und folglich der simultan zu berücksichtigenden Frequenzen besteht.

4.4 Vergleich zum IMSI-Catcher

Praktische Angriffe auf GSM gab es schon bevor der oben beschriebene Angriff durchgeführt wurde. In diesem Zusammenhang wird häufig der Begriff des IMSI-Catchers genannt, der im Folgenden erläutert und von dem bisherigen Angriffsszenario abgegrenzt werden soll. Der erste IMSI-Catcher wurde 1996 von der Deutschen Firma Rohde & Schwarz präsentiert. Seine ursprüngliche Aufgabe bestand in der zwangsweisen Identifikation eines Teilnehmers, in dem man ihn gezwungen hat, seine IMSI (statt der TMSI) zu senden. Mit Hilfe des Netzbetreibers konnte die Telefonnummer (MSISDN) und damit die Identität des Teilnehmers ermittelt werden. Ausgehende Telefongespräche konnten zwar ebenfalls aufgezeichnet werden, stellten aber nicht die Hauptaufgabe des IMSI-Catchers dar. Im Grunde gibt sich der IMSI-Catcher als signalstärkste BTS im Umkreis aus und nutzt die nicht vorhandene Authentifikation des Netzes aus. Gegenüber der MS gibt er vor, keine Verschlüsselung zu unterstützen, während er sich gegenüber dem Netz als gewöhnliches Mobiltelefon ausgibt. Es handelt sich also um einen klassischen Man-in-the-middle Angriff, da der Angreifer in der Lage ist, sowohl die Integrität als auch die Vertraulichket der Kommunikation zu gefährden. Folgenes Sequenzdiagramm (Abbildung 4.5) zeigt den Ablauf eines solchen, aktiven Angriffs. [Str07, S.13ff] Der Angriff mit einem IMSI-Catcher ist also der Klasse der aktiven Angriffe zuzuordnen. Der Einsatz eines sichereren Verschlüsselungsalgorithmus würde keinen Einfluss auf einen solchen Angriff haben. Lediglich eine

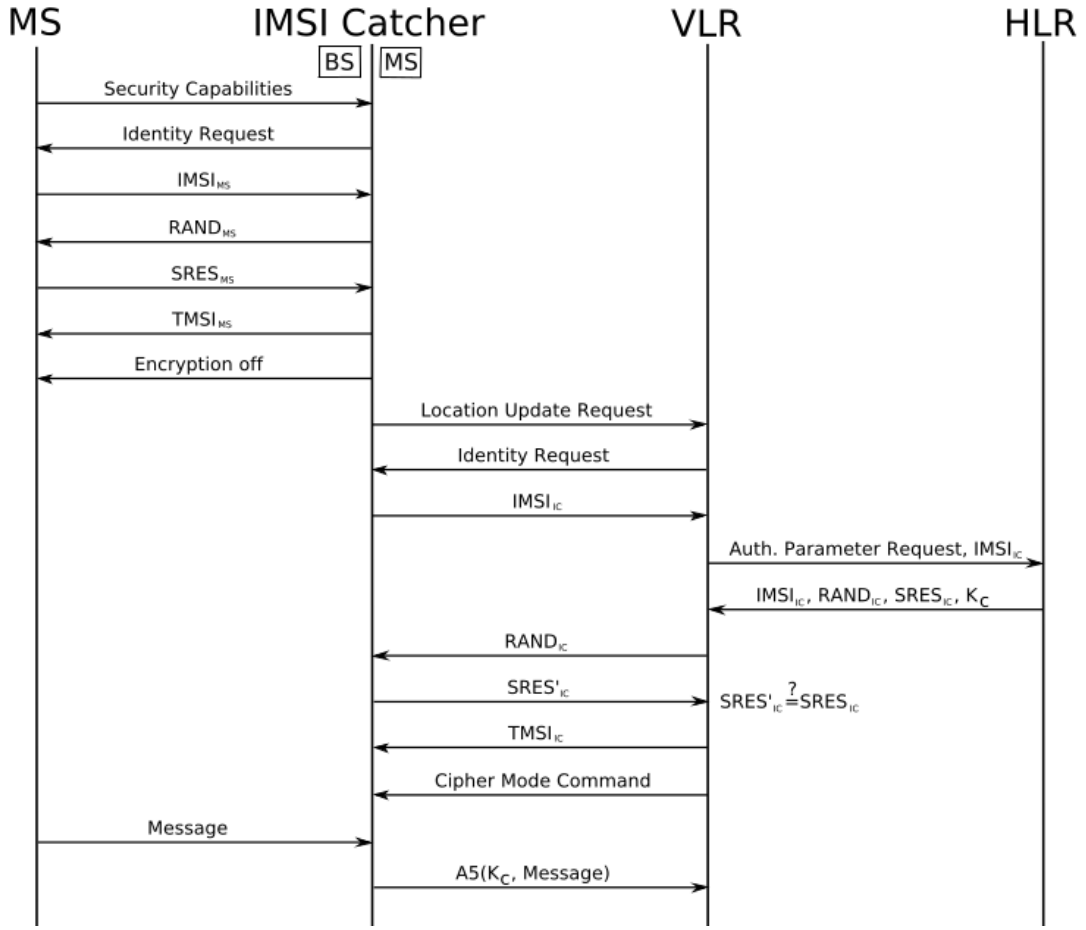

Abbildung 4.5: Angriff mit einem IMSI-Catcher aus [Str07, S.14]

netzseitige Authentifikation und eine Verschlüsselungspflicht wären effektive Maßnahmen zur Abwehr von IMSI-Catchern. Ein IMSI-Catcher kann dank der Open-Source Projekte *OpenBTS*[21] und *OpenBSC*[22] bereits mit vergleichsweise geringem Kostenaufwand [23] emuliert werden. Zur Detektierung von IMSI-Catchern haben Karsten Nohl und Luca Melette im Rahmen eines Patches für OsmocomBB eine Software geschrieben, die den Mobiltelefonbenutzer bei Ortungsversuchen, z.B. einer Silent-SMS alamiert. Das Projekt[24] ist unter dem Namen *Catcher-Catcher* bekannt.[25]

[21] http://wush.net/trac/rangepublic
[22] http://openbsc.osmocom.org/trac/
[23] Im Vergleich zum IMSI-Catcher günstig, jedoch teurer als die oben vorgestellte, passive Methode
[24] http://opensource.srlabs.de/projects/catcher
[25] http://www.tech-blog.net/stille-sms-entdecken-catcher-catcher-macht-es-moglich

5 Schutzmaßnahmen

Nachdem das Sicherheitsrisiko von GSM-Verbindungen sowohl in der Theorie als auch in der Praxis gezeigt wurde, steht der Mobiltelefonbesitzer vor der Frage, welche Schutzmöglichkeiten dem Anwender zur Verfügung stehen. Darüber hinaus stellt sich die Frage, welche Maßnahmen die Provider ergreifen können, um die Sicherheit von GSM zu verbessern. Abschließend soll die Frage geklärt werden, wie sich die aktuelle Sicherheitssituation in Europa, insbesondere in Deutschland darstellt.

5.1 Möglichkeiten der Nutzer

Zunächst sollte sich der GSM-Teilnehmer darüber im Klaren sein, dass die Kommunikation über GSM zum aktuellen Zeitpunkt unsicher ist. Sofern er die Vertraulichkeit für seine Verbindungen gewährleisten möchte, stehen ihm folgende Möglichkeiten zur Verfügung:

- Eine Verschlüsselung auf dem Application Layer z.B. durch Verwendung eines VPN[1] oder auf Hardwareebene direkt am Mikrofon des Telefons wäre eine effektive Möglichkeit zur Sicherstellung der Vertraulichkeit. Diese Methoden setzen jedoch voraus, dass beide Teilnehmer die jeweilige Technik in ihrem Mobiltelefon implementiert haben. Daher erscheint diese Lösung im Allgemeinen als nicht praktikabel.

- Der Teilnehmer kann den Einsatz von Mobiltelefonen bevorzugen, die es erlauben, zwischen GSM und dem Mobilfunkstandard UMTS [2] umzuschalten. Mit dieser Option können nicht nur Datenverbindungen, sondern auch Sprachverbindungen über UMTS gesendet werden, welche einen sichereren Verschlüsselungsalgorithmus als A5/1 verwenden.

- Darüber hinaus könnten Provider bevorzugt werden, die bereits mit der Verbesserung der Sicherheitssituation begonnen haben, beispielsweise mit ersten Umsetzungen der unten aufgeführten Wunschliste. Ein Sicherheitsvergleich unter den Providern ist unten zu finden.

[1] Virtual Private Network
[2] Universal Mobile Telecommunications System

5.2 Wunschliste an die Provider

Karsten Nohl stellte in einem Vortrag über die Sicherheitsproblematik von GSM folgende Wunschliste an die Provider auf [3]:

1. Das *SMS Homerouting* ist eine Routingmethode, welche die SMS auf direktem Weg an den Empfänger sendet und nicht über einen zentralen Kontenpunkt. GSM sendet eine SMS standardmäßig über ein SMSC[4], welches für den Empfang, die Verarbeitung und das Senden von SMS zuständig ist. [Sau08, S.26f] Ein direktes Senden der SMS verbessert nicht die Sicherheitssituation der Luftschnittstelle, schränkt jedoch potentielle Angriffsorte, insbesondere bzgl. Ortung, auf einen lokalen Raum ein. Die 3GPP hat bereits eine Spezifikation für das sogenannte SMS-Routing standardisiert.[5]

2. *Randomized padding:* GSM füllt unvollständige Bursts mit bekannten, statischen Werten auf, die sich im verschlüsselten Zustand auch nicht ändern. Dies ermöglicht einem potentiellen Angreifer eine *Known-Plaintext*-Attacke durchzuführen, die sich verhindern ließe, indem man den Burst mit Zufallswerten statt mit statischen Werten füllt.

3. *Rekeying before each call and SMS:* Aktuell ändert GSM den Sessionkey k_c erst nach mehreren Verbindungen. Ist der Sessionkey berechnet, bzw. gebrochen, kann ein Angreifer auch noch mögliche nachfolgende Gespräche decodieren. Daher ist eine Änderung des Sessionkeys nach jeder Verbindung ein Sicherheitsgewinn und damit eine weitere Forderung an die Provider.

4. *Frequent TMSI changes:* GSM ändert die TMSI eines Teilnehmers ca. ein Mal am Tag. Ist es einem Angreifer gelungen, die TMSI einer Rufnummer bzw. einer Identität zuzuordnen, kann er gezielt Verbindungen aufzeichnen. Daher empfielt es sich, zugunsten der Anonymität des Teilnehmers die TMSI so oft wie möglich, beispielsweise vor bzw. nach jeder Verbindung zu ändern. Zusätzlich kann auch ein Timeout nach einer definierten Zeit während einer Verbindung für eine neue TMSI sorgen.

5. *Frequency hopping* bedeutet, während eines Gesprächs zwischen den Frequenzen zu springen, um einen Angreifer die Aufzeichnung eines ganzen Gesprächs zu erschweren. Diese Forderung haben die meisten Provider bereits umgesetzt, jedoch

[3] Vortrag von Karsten Nohl und Silvian Murnaut (2010): http://events.ccc.de/congress/2010/Fahrplan/attachments/1783_101228.27C3.GSM-Sniffing.Nohl_Munaut.pdf
[4] Short Message Service Center
[5] http://www.3gpp.org/ftp/Specs/html-info/23840.htm

besteht die Forderung, die Frequenzsprünge häufiger und zufälliger durchzuführen. Diese Forderung verbessert die Sicherheit m.E. nur sehr marginal. Frequenzsprünge können nachträglich leicht nachvollzogen werden, sofern ein entsprechend großes Frequnzspektrum aufgezeichnet wurde.

Sämtliche Punkte auf der Wunschliste sind m.E. aus Sicht der Providers leicht umzusetzen. Folgende, etwas schwieriger umzusetzende Forderungen können der Wuschliste hinzugefügt werden:

- Der aktuell verwendete Standard A5/1 kann mittlerweile mit einfachen Mitteln in kurzer Zeit gebrochen werden. Daher wäre es m.E. ratsam, auf einen sichereren Standard (z.B. A5/4) zu wechseln. Da die Verschlüsselung in jeder BTS größtenteils in Hardware implementiert ist, wäre eine Erneuerung bzw. Verbesserung mit hohen Kosten für die Netzbetreiber verbunden. Das Mobiltelefon muss im Rahmen dieser Maßnahme ebenfalls A5/3 bzw. A5/4 unterstützen, was bei den neueren Modellen jedoch bereits gegeben ist.

- Der Schutz vor aktiven Eingriffen in die Luftschnittstelle ist nicht Thema dieser Arbeit, soll aber kurz in Form einer m.E. effektiven Schutzmaßnahme genannt werden. In GSM authentifiziert sich lediglich die MS gegenüber dem Netz, jedoch findet eine netzseitige Authentifikation nicht statt. Die Annahme, das Netz sei vertrauenswürdig, gilt, obwohl aktive Angriffe existieren, die Gegenteiliges beweisen. Eine netzseitige Authentifikation verhindert keine passiven, dafür sämliche aktive Angriffe, wie z.B. die Emulation einer BTS oder den Einsatz von nicht amtlich genehmigten IMSI-Catchern. Ein Modell ähnlich dem asymetrischen Verschlüsselungs- und Authentifikationsverfahren PGP[6] könnte bei GSM für eine sichere Verschlüsselung und Authentifikation sorgen. Der private Schlüssl könnte beispielsweise an Stelle des jetzigen k_i treten.

5.2.1 Sicherheitsvergleich europäischer Staaten und Provider

Der aktuelle Sicherheitsgrad von Staaten und Providern wird von der Firma *Security Research Labs GmbH*[7](kurz: *srlabs*) auf der Webseite gsmmap.org in Form einer Weltkarte zusammengetragen und ausgewertet. Da es sich um ein Open-Source Projekt handelt, können die Rohdaten dezentral und von jeder Person, die sich an diesem Projekt beteiligen möchte, gesammelt werden. Obgleich die Datengranularität in Europa deutlich feiner ist,

[6]Pretty Good Privacy
[7]Managing director: Karsten Nohl

gilt die dortige Karte für alle Staaten der Erde, die GSM-Netze betreiben. Dabei wird die Sicherheit des GSM-Netzes anhand folgender Kategorien[8], welche die obige Wunschliste zum Teil abstrahieren gemessen:

Abfangen (Intercept)

Ist es möglich Textnachrichten oder Anrufe aufzuzeichnen?

Imitation (Impersonation)

Kann ein Angreifer die Identität eines Teilnehmers missbrauchen, um sich z.B. einen finanziellen Vorteil zu verschaffen?

Lokalisierung (Tracking)

Kann die Position eines Teilnehmers anhand öffentlich zugänglicher Informationen ermittelt werden?

Der öffentliche Vergleich hat die Intention, den Druck auf die Betreiber zu erhöhen, um eine Veränderung bzgl. der Sicherheit der Kommunikationsinfrastruktur zu gewährleisten.[9] Den europäischen Sicherheitsvergleich in Bezug auf das Abfangen von GSM-Daten zeigt folgende Karte[10] (Abbildung 5.1).

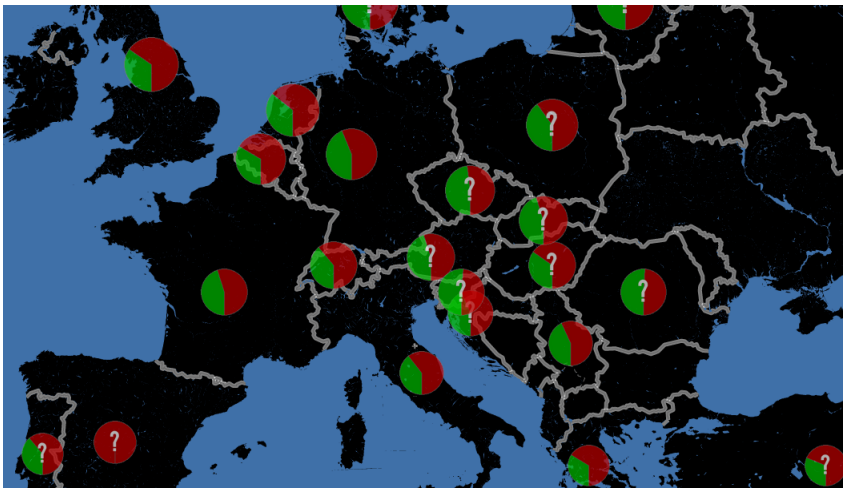

Abbildung 5.1: GSM-Sicherheit bzgl. Tracking in Europa 2012

Die staatenspezifische Sicherheitsstatistik lässt sich auch feingranularer auf die einzelnen Provider abbilden. Abbildung 5.2 [11] zeigt die Situation in Deutschland, bezogen auf die Provider O2, E-Plus, Vodafone und T-Mobile. Der Grafik ist zu entnehmen, dass die

[8] https://srlabs.de/gsmmap/
[9] https://srlabs.de/gsmmap/
[10] Ausschnitt aus http://gsmmap.org/
[11] Ausschnitt aus http://gsmmap.org/

Sicherheitssituation bezüglich des Abfangens von GSM-Daten bei T-Mobile aktuell im Vergleich zu den anderen deutschen Providern am besten ist. Der Idealfall definiert sich dabei als ein GSM-Netz, welches alle Sicherheitstechniken verwendet, die jemals ein GSM-Netz auf der Welt implementiert hat.

Abbildung 5.2: GSM-Sicherheit bzgl. Tracking in Deutschland 2012

Abschließend sei jeder GSM-Teilnehmer aufgerufen, sich an der Vervollständigung und Aktualisierung der GSM-Daten zu beteiligen. Srlabs stellt zu diesem Zweck ein spezielles Tutorial[12] bereit. Vorraussetzung ist ein Osmocom-kompatibles Mobiltelefon [13].

[12]https://srlabs.de/gsmmap-liveiso-tutorial
[13]Eine Liste kompatibler Mobiltelefone ist hier zu finden: http://bb.osmocom.org/trac/#SupportedPhonehardware

6 Fazit

GSM hat sich als eine unsichere Kommunikationstechnologie herausgestellt, der seitens der Mobilfunknutzer und Provider vertraut wird. Es wurde gezeigt, dass es trotz geringem Kostenaufwand möglich ist, die Privatsphäre anderer Menschen zu verletzen.

Die Schutzmöglichkeiten der Nutzer sind aufgrund der Netzdominanz stark eingeschränkt und für den weniger kundigen Nutzer oft unpraktikabel. Daher sind die Provider bzw. Netzbetreiber aufgerufen die bestehende Situation zu verbessern. Die flächendeckende Substitution von GSM durch modernere Standards, wie z.B. UMTS, wäre möglicherweise kostengünstig, jedoch m.E. aufgrund der unterschiedlichen Anwendungsgebiete nicht sinnvoll. Vielmehr sollte in die Sicherheit von GSM investiert werden, wobei sich einige Defizite mit geringem Kostenaufwand ausgleichen lassen. Die Investition in neue Hardware zur netzseitigen Unterstützung neuerer Verschlüsselungsmethoden ist dennoch m.E. unverzichtbar.

Mit dieser Arbeit soll das Ziel erfüllt sein, einen Beitrag zur Aufklärung über die aktuelle Sicherheitssituation der GSM Luftschnittstelle geleistet zu haben, um darüber hinaus den Druck auf die Provider bzw. Netzbetreiber anteilig zu erhöhen. Ferner besteht die Hoffnung, die Privatsphäre aller Nutzer partiell zu verbessern und ein Bewusstsein für sichere Kommunikation über den Mobilfunk hinaus geschaffen zu haben.

Literaturverzeichnis

[3GP00] 3GPP TSG SA WG3 Security: GSM Association Specification for A5/3. In: *GSMA TSG SA WG3 Working party* (2000). http://www.3gpp.org/ftp/tsg_sa/wg3_security/tsgs3_13_yokohama/docs/pdf/s3-000362.pdf

[3rd09] 3rd Generation Partnership Project: Specification of the A5/4 Encryption Algorithms for GSM. In: *3GPP TS* (2009). http://serving.webgen.gsm.org/5926DA9A-2DD6-48E7-BAD4-50D4CD3AF30A/assets/a54specification.pdf

[Bet11] Betz, Johann: *Analyse und praktische Umsetzung der Entschlüsselung von GSM-Telefonaten.* http://www.data.ks.uni-freiburg.de/download/masterarbeit/SS11/09-betz-gsm/abschlusspraes.pdf. Version: September 2011

[Dia12] Diaconescu, Bogdan: *Evaluating GSM A5/1 security on hopping channels.* http://yo3iiu.ro/blog/wp-content/uploads/2012/04/Evaluating_GSM_hopping1.pdf. Version: April 2012

[Ger12] Gerhard, Marc Dimi: *Grundlagen zu GSM.* https://image.informatik.htw-aalen.de/Karg/Labor-IT-Sicherheit/Docs/gsm-skript.pdf. Version: Mai 2012

[Gö03] Göller, Dr.-Ing. J.: *Die GSM-Dm-Kanäle im Dialog.* Duderstadt : EPV Elektronik-Praktiker-Verlagsgesellschaft mbH, 2003. – ISBN 978–3–936318–00–5

[Hei99] Heine, Gunnar: *GSM Networks: Protocols, Terminology, and Implementation.* Boston, London : Artech House, 1999. – ISBN 0–89006–471–7

[Kow02] Kowalk, Prof. Dr. W.: *Rechnernetze Onlineskript, GSM.* http://einstein.informatik.uni-oldenburg.de/rechnernetze/gsm.htm. Version: März 2002

[Mey10] Meyer, Steven: *Breaking GSM with rainbow Tables.* http://arxiv.org/pdf/1107.1086.pdf. Version: März 2010

[Oec03] Oechslin, Philippe: *Making a Faster Cryptanalytic Time-Memory Trade-Off.* http://lasecwww.epfl.ch/pub/lasec/doc/Oech03.pdf. Version: Mai 2003

[Oy02] Oy, Nokia N.: GSM Air Interface & Network Planning - Training Document. In: *TC Finland Issue* (2002). http://www.roggeweck.net/uploads/media/Student_-_GSM_Air_Interface___NW_Planning.pdf

[Qui04] Quirke, Jeremy: Security in the GSM system. In: *AusMobile* (2004). http://web.archive.org/web/20040712061808/www.ausmobile.com/downloads/technical/Security+in+the+GSM+system+01052004.pdf

[Sau08] Sauter, Martin: *Grundkurs Mobile Kommunikationssysteme - Von UMTS und HSDPA, GSM und GPRS zu Wireless LAN und Bluetooth Piconetzen.* 3. erweiterte Auflage. Wiesbaden : Friedr. Vieweg & Sohn Verlag, 2008. – ISBN 978–3–8348–0397–9

[Str07] Strobel, Daehyun: *IMSI Catcher.* http://www.emsec.rub.de/media/crypto/attachments/files/2011/04/imsi_catcher.pdf. Version: Juli 2007

[Sü03] Südmeyer, Philipp: *Die Stromchiffre A5.* www.suedmeyer.net/inhalte/pdf/a5_thesis.pdf. Version: November 2003

[Sü06] Südmeyer, Philipp: *How Rainbow Tables work.* http://kestas.kuliukas.com/RainbowTables/. Version: Dezember 2006

[Wel12] Welte, Harald u.a.: *OsmocomBB Projekt.* http://bb.osmocom.org/trac/. Version: Juni 2012

Sämtliche Links wurden zuletzt am 16. Oktober 2012 besucht und bezüglich ihrer Erreichbarkeit geprüft.

Sachregister

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschatlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.

Oldenburg, den 16. Oktober 2012

Peter Gewald