

# Ein Echtzeitverschlüsselungssystem für Mikrofonfunkstrecken

## Bachelor-Thesis

zur Erlangung des akademischen Grades B.Sc.

**Antonia Schwab**



Hochschule für Angewandte Wissenschaften Hamburg  
Fakultät Design, Medien und Information  
Department Medientechnik

Erstprüfer: Prof. Eva Wilk

Zweitprüfer: Prof. Jan Mietzner

10. Juli 2019

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Ziel und Motivation . . . . .	5
1.1.1	Verschlüsselung des Signals . . . . .	5
1.1.2	Echtzeitübertragung . . . . .	5
1.1.3	Systemaufbau . . . . .	6
<b>2</b>	<b>Planung</b>	<b>7</b>
2.1	Kompatibilität . . . . .	7
2.2	Kryptographie-Verfahren . . . . .	7
2.2.1	Kryptographische Sicherheit . . . . .	8
2.2.2	Substitutionsverfahren . . . . .	8
2.2.3	Transpositionsverfahren . . . . .	9
2.2.4	Erwartete Latenzen für unterschiedliche Blocklängen . . . . .	15
2.3	Synchronisierung von Sender und Empfänger . . . . .	16
2.3.1	Barker-Codesequenzen . . . . .	16
2.3.2	Unterschiedliche Abtastraten . . . . .	16
2.4	Kanaleinfluss . . . . .	19
2.4.1	One Time Pad über Kanal . . . . .	20
2.4.2	Scrambling über Kanal . . . . .	21
2.4.3	Kanaleinfluss auf Barker-Codesequenz . . . . .	22
2.4.4	Gesamtsignal über Kanal . . . . .	23
2.5	Signale und Pegel an Übergangspunkten . . . . .	25
2.5.1	Chiffrierer . . . . .	25
2.5.2	Dechiffrierer . . . . .	26
2.6	Mobilität und Nutzbarkeit . . . . .	26
<b>3</b>	<b>Umsetzung</b>	<b>28</b>
3.1	Wahl der Plattform . . . . .	28
3.2	Analoge Schaltungen . . . . .	28
3.2.1	Chiffrierer . . . . .	29
3.2.2	Dechiffrierer . . . . .	30
3.2.3	Anschlüsse und Impedanzen . . . . .	31
3.3	Programmierung . . . . .	32
3.3.1	Programmstruktur . . . . .	32
3.3.2	Synchronisierung . . . . .	34
3.3.3	Chiffrierung und Dechiffrierung . . . . .	36

## *Inhaltsverzeichnis*

3.4	Probleme und Verbesserungsmöglichkeiten . . . . .	39
<b>4</b>	<b>Validierung</b>	<b>44</b>
4.1	Durchzuführende Messungen . . . . .	44
4.2	Amplitudenfrequenzgang . . . . .	45
4.3	Phasenfrequenzgang . . . . .	47
4.3.1	Phasenfrequenzgang PGX-System . . . . .	47
4.3.2	Phasenfrequenzgänge Sender, Empfänger, beide . . . . .	48
4.3.3	Messungen mit Verschlüsselung . . . . .	50
4.4	Klirrfaktor . . . . .	59
4.4.1	Klirrfaktor bei unterschiedlichen Schlüsseln . . . . .	61
4.4.2	Bewertung der Ergebnisse . . . . .	63
4.5	Messung des Gesamtsystems mit Verschlüsselung . . . . .	64
4.5.1	Barker-Codesequenz über das System . . . . .	64
4.5.2	Bewertung der Ergebnisse . . . . .	67
4.6	Hörtest zur Validierung der Verschlüsselung . . . . .	68
4.6.1	Aufbau und Zielsetzung . . . . .	68
4.6.2	Ergebnisse und Bewertung . . . . .	69
<b>5</b>	<b>Zusammenfassung</b>	<b>72</b>
<b>A</b>	<b>Material</b>	<b>74</b>
A.1	Beigefügte CD . . . . .	74
A.2	Liste über die Benennung und Inhalt der Audiodateien . . . . .	74
A.3	Analoge Schaltungen des Chiffrierers und Dechiffrierers . . . . .	75
A.4	Programmablaufpläne der Sketche Chiffrierer und Dechiffrierer . . . . .	78
A.5	Fragebögen und Notizen zum Hörtest . . . . .	81
A.6	Auskunft zu Mikrofonfunkstrecken der Firmen Shure und Beyerdynamic per E-Mail . . . . .	88
	<b>Abbildungsverzeichnis</b>	<b>92</b>
	<b>Tabellenverzeichnis</b>	<b>95</b>
	<b>Literaturverzeichnis</b>	<b>96</b>

## **Abstract**

Nowadays it is increasingly important to create an environment protecting against wiretapping while using wireless microphones. This thesis proposes a prototypical real time encryption system for analog wireless microphones that do not have an inherent encryption method. The system is implemented on two Arduino Due to examine to what extent the system is viable. A digital encryption, that uses a transposition method to swap samples, is used. The transposition block length varies depending on a key. The validation shows that the system has limited usability because of missing bit synchronization.

## **Zusammenfassung**

In der heutigen Zeit wird es zunehmend wichtiger bei der Verwendung von Mikrofonfunkstrecken eine abhörsichere Umgebung zu schaffen. Damit analoge Mikrofonfunkstrecken, die keine Verschlüsselungsmöglichkeit bieten, weiter verwendet werden können, wird in dieser Arbeit ein prototypisches Echtzeitverschlüsselungssystem zur Ergänzung der Mikrofonfunkstrecke entworfen. Dieses wird anschließend auf zwei Arduino Due umgesetzt, um zu prüfen in welchem Umfang das System funktionsfähig ist. Verwendet wird eine digitale Verschlüsselung, bei der mittels eines Transpositionsverfahrens Abtastwerte miteinander vertauscht werden. Je nach verwendetem Schlüssel sind die Vertauschungsblocklängen unterschiedlich. Die Validierung zeigt allerdings, dass das System in Kombination mit Mikrofonfunkstrecken aufgrund der fehlenden Bitsynchronisation nur sehr eingeschränkt nutzbar ist, obwohl die Ziele der Echtzeit und der Sprachverständlichkeit erreicht werden können.

# 1 Einleitung

## 1.1 Ziel und Motivation

Heutzutage wird es zunehmend wichtiger, Daten und Signale zu verschlüsseln und dadurch die Vertraulichkeit von Gesprächen zu schützen. Dies schließt die Übertragung von Audiosignalen über Funkstrecken mit ein. Allerdings ist eine sichere Audiosignalverschlüsselung bisher nur mit digital arbeitenden Systemen möglich. Analoge Mikrofonfunkstrecken bieten hingegen keine Verschlüsselung an. Somit können diese Systeme nicht verwendet werden, wenn eine abhörsichere Umgebung gewünscht ist und müssen gegebenenfalls durch digitale Funkstrecken mit Verschlüsselungsmöglichkeit ersetzt werden. Da dies oft aufwändige Umbauarbeiten und hohe Kosten verursacht, wäre es denkbar, das vorhandene System zu erweitern, sodass es damit möglich wird, Sprachsignale zu verschlüsseln. Dieses Verschlüsselungssystem soll in dieser Arbeit prototypisch entwickelt werden. Ziel dabei ist es, dass die Echtzeit des Systems durch die Verschlüsselung erhalten bleibt. Die Verschlüsselung hat nicht den Anspruch, gegen Hackerangriffe absolut sicher zu sein. Das Ziel ist jedoch, dass das verschlüsselte Signal nicht mehr verständlich ist.

Zunächst werden daher in Frage kommende Verschlüsselungsverfahren betrachtet und geprüft, ob diese für die geplante Verwendung in Kombination mit einer analogen Mikrofonfunkstrecke theoretisch funktionieren können. Das entworfene System wird anschließend praktisch umgesetzt. Es folgt eine messtechnische Überprüfung unter Verwendung einer ausgewählten Mikrofonfunkstrecke in Hinblick auf die Echtzeit und die Audioqualität. Außerdem wird mittels eines Hörversuches überprüft, ob das verschlüsselte Signal bei unbefugtem Mithören zu verstehen ist.

### 1.1.1 Verschlüsselung des Signals

Die Verschlüsselung soll digital erfolgen, sodass die Grenzen eines digitalen Verschlüsselungssystems mit einer analogen Funkübertragungsstrecke untersucht werden können. Denkbar wäre auch, der Verschlüsselung eine Kanalcodierung nachzuschalten, um mögliche Fehler bei der Übertragung erkennen und korrigieren zu können. Dies ist aber im Rahmen dieser Arbeit bewusst nicht geplant. Auch ein sicherer Schlüsselaustausch ist für das geplante System zunächst nicht priorisiert.

### 1.1.2 Echtzeitübertragung

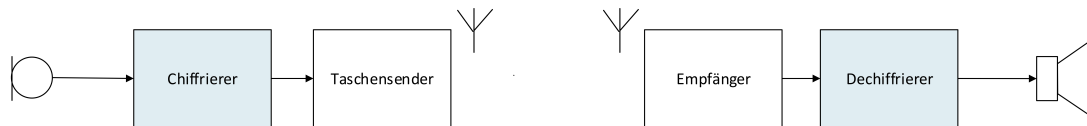
Das Ziel bei der Entwicklung des Systems ist die Erhaltung der Echtzeit.

## 1 Einleitung

„'Echt-Zeit' setzt sich zusammen aus der 'Zeit' als physikalische Größe der Datenverarbeitung und 'Echt' im Sinne von 'für den Menschen als real empfundene Zeit'.“ (Gessler 2014: 27).

Um eine Beurteilung darüber machen zu können, ob das System in Echtzeit funktioniert, muss daher definiert werden, welcher physikalisch messbare Wert für die wahrgenommene Zeit der Audioübertragung gilt. Das menschliche Ohr nimmt zwei identische Signale, die zeitverzögert gesendet werden, erst ab einer Verzögerungszeit in Höhe von 100 ms als Echo wahr. Ein Verwischungseffekt tritt jedoch schon ab einer Verzögerungszeit von 50 ms auf ((Baunetz Wissen 2015)). Das Ziel ist jedoch eine Verzögerungszeit von nicht mehr als 100 ms über die gesamte Signalkette zu erreichen.

### 1.1.3 Systemaufbau



**Abbildung 1.1:** Schematische Darstellung des geplanten Systemaufbaus

Das Mikrofonfunksystem soll weiterhin die Aufgabe der Funkübertragung übernehmen und ist in Abbildung 1.1 weiß dargestellt. Die blau gefärbte Verschlüsselungseinheit soll zwischen Mikrofon und Funksender in die Signalkette eingefügt werden. Dies führt zu einer systembedingten Einschränkung, da bei Handsendern Mikrofon und Sender physikalisch in einem Gerät ausgeführt sind. Die Funksendeeinheit soll aber weder verändert noch auseinander gebaut werden müssen, daher beschränkt sich das geplante Verschlüsselungssystem auf die Nutzung mit einem Taschensender in Kombination mit einem beliebigen Mikrofon. Der Einsatz mit einem Handmikrofon ist somit nicht möglich.

## 2 Planung

Für die Planung des Systems werden bereits aufgezeichnete Testsignale verwendet, sodass in dieser Phase noch nicht in Echtzeit gearbeitet wird. Mit Hilfe des Programmes Matlab wird im Vorfeld die verwendete Verschlüsselung ermittelt und geprüft, ob diese dem Einfluss eines realen Kanals standhält.

### 2.1 Kompatibilität

Im Idealfall wäre das Verschlüsselungssystem zu allen Systemen, die es heutzutage am Markt gibt, kompatibel. Mikrofonfunkstrecken, die digitale Modulationsverfahren wie FSK-Verfahren (zum Beispiel das T1000-System von Beyerdynamic ([Gmoser 2019](#))) oder 4-PSK (QLX- und ULX-Systeme von Shure) beziehungsweise 16-QAM (Axient Digital von Shure ([Schwörer 2019](#))) nutzen, könnten theoretisch mit dem geplanten Verschlüsselungssystem kompatibel sein. Allerdings haben diese Systeme ohnehin die Möglichkeit, das Audiosignal verschlüsselt zu übertragen, somit wäre es unnötig das Verschlüsselungssystem in Kombination mit diesen Systemen zu verwenden.

Die eingangs erwähnte Einschränkung bei der Verwendung von Handsendern gilt ebenso für Sender, die als Grenzflächenmikrofon ausgeführt sind. Die Verwendung von Sprechstellen mit einem wechselbaren Schwanenhalsmikrofon wäre aber theoretisch möglich. Im Rahmen dieser Arbeit wird allerdings nur auf die Verwendung mit einem Taschensender eingegangen. Da es jedoch nicht möglich ist, alle auf dem Markt verfügbaren analogen Funkstrecken zu testen, wird das System beispielhaft mit dem PGX-System von Shure ([Shure 2010](#)), das aus dem Empfänger PGX4 und dem Taschensender PGX1 besteht, getestet. Als Mikrofon wird ein SM58 ([Shure 2014](#)), ebenfalls von Shure, verwendet. Ob auch andere Systeme mit dem geplanten Verschlüsselungssystem kompatibel sind, hängt von der Übertragungsbandbreite, Eingangs- und Ausgangspegeln und eventuell verbauten Kommandersystemen (4.5) ab. Das verwendete Frequenzband zur Funkübertragung sollte theoretisch keinen Einfluss auf die Kompatibilität haben.

### 2.2 Kryptographie-Verfahren

Durch die Verschlüsselung wird ein Klartext, in diesem Fall ein Sprachsignal, mittels eines Algorithmus in einen Schlüsseltext beziehungsweise ein verschlüsseltes Signal gewandelt.

Man unterscheidet in der Kryptographie zwischen Substitutionsverfahren und Transpositionsverfahren. Bei Substitutionsverfahren wird der Wert jedes Symbols durch einen anderen Wert ersetzt, behält aber seine Position. Im Gegensatz dazu wird bei Transposition die Position der Symbole getauscht, der einzelne Wert jedes Symbols bleibt aber erhalten. (Meyer 2014: 266)

Moderne Kryptographie-Verfahren, wie der AES-Algorithmus, nutzen eine Kombination aus beiden Verfahren (Ertel & Löhmann 2018: 69ff.). Mikrofonfunkstrecken, die eine Verschlüsselung anbieten, verwenden ebenfalls den AES-Algorithmus, jedoch mit unterschiedlicher Schlüssellänge (Schwörer 2019, Gmoser 2019).

### 2.2.1 Kryptographische Sicherheit

Ein Angreifer wählt immer die für ihn einfachste Methode, ein Verschlüsselungssystem zu knacken, daher muss es *„auch dann noch sicher sein, wenn der Angreifer alle Details des Kryptosystems kennt, mit der Ausnahme des Schlüssels.“* (Paar & Pelzl 2016: 11) So sorgen verlorene oder gestohlene Systeme nicht dafür, dass andere Systeme dieser Art nie mehr verwendet werden könnten. Fehlt dem Angreifer also nur das Wissen über den richtigen Schlüssel, gibt es für ihn mehrere Möglichkeiten, diesen herauszufinden.

Zum Einen kann er alle möglichen Schlüsselkombinationen ausprobieren. Hierbei gilt ein System als *„informationstheoretisch [...] sicher, wenn es auch dann nicht gebrochen werden kann, wenn dem Angreifer beliebige Rechenleistung zur Verfügung steht.“* (Paar & Pelzl 2016: 41) Allerdings können auch *„zwischenmenschliche Beeinflussungen [...] ausgenutzt [werden], um Kryptoschlüssel zu erhalten.“* (Paar & Pelzl 2016: 11) Diese Art der Angriffe kann verhindert werden, wenn das System den Schlüssel selbst generiert und ihn weder der Nutzer noch der Entwickler kennt.

Auch der Schlüsseltausch stellt gegebenenfalls ein Sicherheitsrisiko dar. Bei Mikrofonfunkstrecken, die eine Verschlüsselung anbieten, erfolgt der Schlüsselaustausch bei der Synchronisierung von Sender und Empfänger über die Infrarot-Schnittstelle. (Schwörer 2019, Gmoser 2019)

### 2.2.2 Substitutionsverfahren

#### One Time Pad

Ein Beispiel für ein theoretisch perfektes und nicht entschlüsselbares Verfahren ist das One Time Pad. (Paar & Pelzl 2016: 41) Es nutzt für die Verschlüsselung eine XOR-Operation der Nachricht mit dem Schlüssel, wobei der Schlüssel genauso lang ist wie die zu verschlüsselnde Nachricht selbst (Beutelspacher 2015: 60). Für die Entschlüsselung erfolgt eine weitere XOR-Operation des verschlüsselten Signals mit dem Schlüssel. Damit das One Time Pad perfekte Sicherheit bietet, ist es wichtig, dass der Schlüssel aus einer perfekt zufälligen Bitfolge bestehen. (Ertel & Löhmann 2018: 53 ff.) Da jedoch die Übermittlung eines unendlich langen Schlüssels nicht zweckmäßig



ist, bietet es sich an, mit Hilfe eines Schieberegisters eine endlich lange pseudozufällige Bitfolge zu realisieren. (Beutelspacher 2015: 63) Das Schieberegister könnte zu Beginn der Verwendung jeweils einen neuen Schlüssel generieren, der für die Dauer des Betriebs dann verwendet wird.

Somit würde das System zwar keine informationstheoretische perfekte Sicherheit mehr bieten, es wäre allerdings für die Betriebsdauer immer noch ausreichend sicher. Eine Schlüssellänge von nur 56-64 Bit bietet bereits eine kurzfristige Sicherheit von einigen Stunden oder Tagen (Paar & Pelzl 2016: 13). Allerdings hat das One Time Pad bei der Verwendung mit Sprachsignalen den Nachteil, dass das Signal nach der Chiffrierung immer noch zu verstehen ist. (Audiofile F3-OTP65536) Um dieses Problem zu lösen, wäre es denkbar, dem Signal ein pseudozufälliges Rauschen zu überlagern, das sowohl auf Sender- als auch Empfängerseite bekannt ist und so wieder entfernt werden kann. Auch Lösungen wie die Kombination mit einem Transpositionsverfahren (2.2.3) wären denkbar. Es gibt aber ein noch größeres Problem, das durch die Übertragung über einen analogen Kanal entsteht (2.4.1). Daher wird für die Umsetzung ein anderes Verschlüsselungsverfahren gewählt.

### 2.2.3 Transpositionsverfahren

Versetzungsverfahren verändern die Reihenfolge der zu verschlüsselnden Symbole. Der Wert der einzelnen Symbole bleibt dabei erhalten. Das Schema der Transposition kann beliebig sein, häufig wird dafür jedoch eine zweidimensionale Matrix verwendet. (Wätjen 2018: 15)

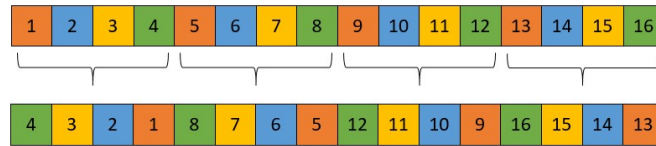
Es handelt sich bei Transpositionsverfahren um ein symmetrisches Verfahren, da der Schlüssel, der die Art der Transposition bestimmt, auf Sender- und Empfängerseite der selbe ist. (Meyer 2014: 266)

#### Mögliche Umsetzung eines Transpositionsverfahrens

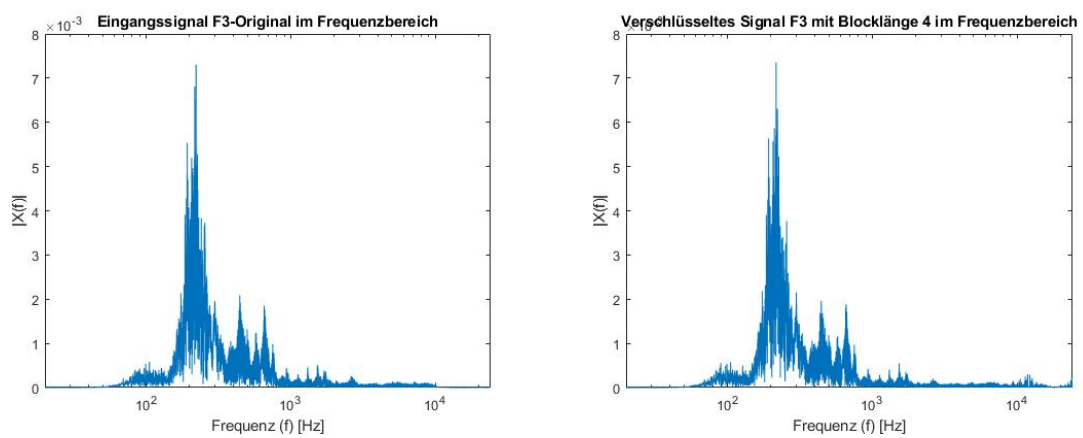
Das zu verschlüsselnde Sprachsignal besteht nach der Analog-Digital-Wandlung nun aus einzelnen Abtastwerten, die je nach Quantisierung jeweils aus einer bestimmten Anzahl an Bits bestehen. Es können nun die gesamten Abtastwerte, die jeweils je nach Bittiefe aus unterschiedlich vielen Bits bestehen, miteinander vertauscht werden. Denkbar wäre allerdings auch das Vertauschen der einzelnen Bits. Dies könnte entweder innerhalb eines Abtastwertes erfolgen oder die Bits mehrere Abtastwerte mit einschließen. In einem einfachen Versuch werden zunächst die ersten vier Abtastwerte in eine Matrix gespeichert und anschließend wieder ausgegeben, wobei sich die Reihenfolge der vier Abtastwerte jeweils umkehrt wird (siehe Abbildung 2.1).

Es ist allerdings festzustellen, dass die Anzahl von vier Werten lange nicht ausreicht, um das Sprachsignal unverständlich zu machen. Betrachtet man das Spektrum des Originalsignals im Vergleich mit dem verschlüsselten Signal, wird dies sichtbar. Es sind kaum Unterschiede zu erkennen, lediglich enthält das verschlüsselte im Gegensatz zum Originalsignal Frequenzen über 10kHz. Dies ist auf den entsprechenden Audi-

## 2 Planung



**Abbildung 2.1:** Scramblingstruktur bei einer Blocklänge von 4 Werten

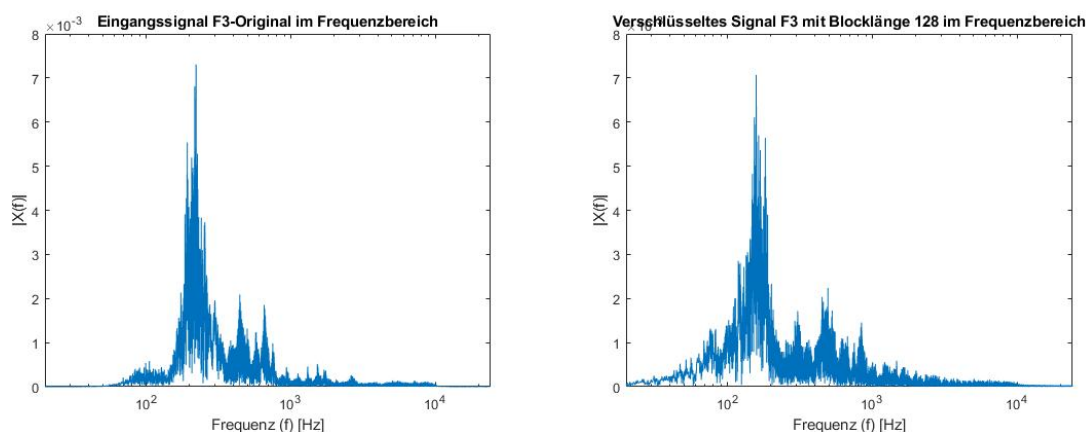


**Abbildung 2.2:** Vergleich der Spektren des Originalsignals F3 mit der Verschlüsselung des Signals mit Blocklänge 4

## 2 Planung

odateien F3-Original und F3-4-Matlab gut zu hören. Ist die Abtastrate im Vergleich zur Blocklänge innerhalb derer die Werte vertauscht werden hoch, ist noch keine Verschlüsselung gegeben. Durch den hohen deterministischen Anteil eines Sprachsignals sind zwischen benachbarten Abtastwerten keine großen Änderungen zu erwarten und somit führt eine Vertauschung von wenigen Werten auch noch nicht dazu, dass Sprachverständlichkeit erreicht wird.

Verlängert man nun die Blocklänge auf 128 Werte, sieht der Vergleich der Spektren des Originalsignals mit dem verschlüsselten Signal folgendermaßen aus: Es ist immer



**Abbildung 2.3:** Vergleich der Spektren des Originalsignals F3 mit dem der Verschlüsselung des Signals mit Blocklänge 128

noch kein deutlicher Unterschied zwischen beiden Spektren zu erkennen, allerdings weist das verschlüsselte Signal nun auch tiefe Frequenzen unterhalb von 60 Hz auf und der Pegel des Bereichs von 60 Hz bis circa 150 Hz ist im Vergleich zum Originalsignal nun höher. Auch zu erkennen ist, dass sich die Hauptfrequenz, die im Originalsignal bei 221 Hz liegt nun verschoben hat und nach der Verschlüsselung bei 158 Hz liegt. Das Sprachsignal ist allerdings trotz der nun hörbaren Verzerrungen und einem erhöhten Bassbereich noch gut zu verstehen (Audiofile F3-128-Matlab-rückwärts).

Selbst bei einer Blocklänge von 4096 Abtastwerten, lassen sich einzelne Sprachschnipsel verstehen (Audiofile F3-4096-Matlab). In Hinblick auf die Latenz ist allerdings eine möglichst kleine Blocklänge von Vorteil (siehe Kapitel 2.1). Daher werden die Abtastwerte innerhalb der Blocklänge noch weiter verschachtelt.

Die 128 Werte werden in eine 8x16-Matrix geschrieben (blauer Pfeil in Abbildung 2.4). Die Werte werden nun beginnend beim roten Pfeil ausgegeben. Nach Wert 8 folgt 127, dann 119 und so weiter. Durch diese Umstrukturierung lässt sich das Sprachsignal nur noch schwer verstehen, selbst wenn der Inhalt beispielsweise als Text gleichzeitig zu lesen ist und daher schon bekannt ist. (Audiofile F3-128-Matlab). Auch der Vergleich der Spektren des Originalsignals mit dem nach diesem Schema verschlüsselten Signal zeigt nun deutlich größere Unterschiede.

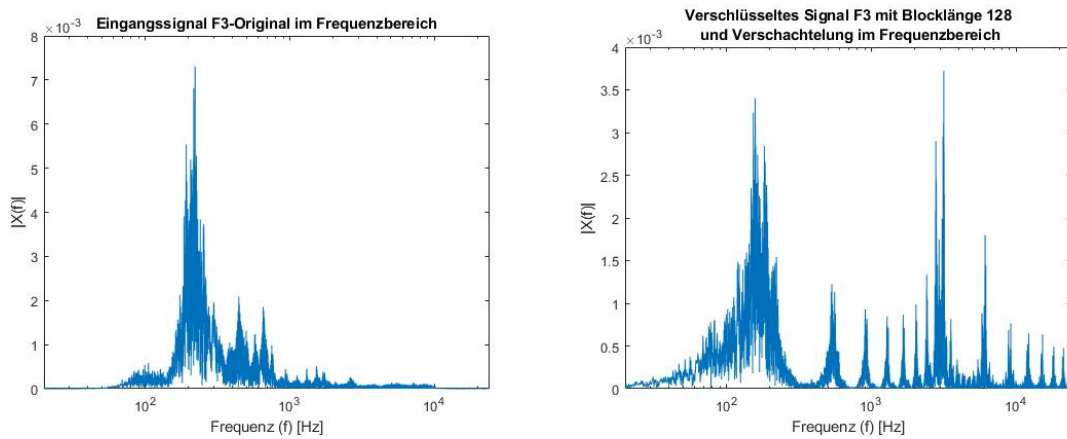
Darstellung der Verschlüsselungsstruktur

↓

1	9	17	25	33	41	49	57	65	73	81	89	97	105	113	121
2	10	18	26	34	42	50	58	66	74	82	90	98	106	114	122
3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123
4	12	20	28	36	44	52	60	68	76	84	92	100	108	116	124
5	13	21	29	37	45	53	61	69	77	85	93	101	109	117	125
6	14	22	30	38	46	54	62	70	78	86	94	102	110	118	126
7	15	23	31	39	47	55	63	71	79	87	95	103	111	119	127
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128

←

**Abbildung 2.4:** Darstellung der Verschlüsselungsstruktur bei einer Blocklänge von 128 Werten



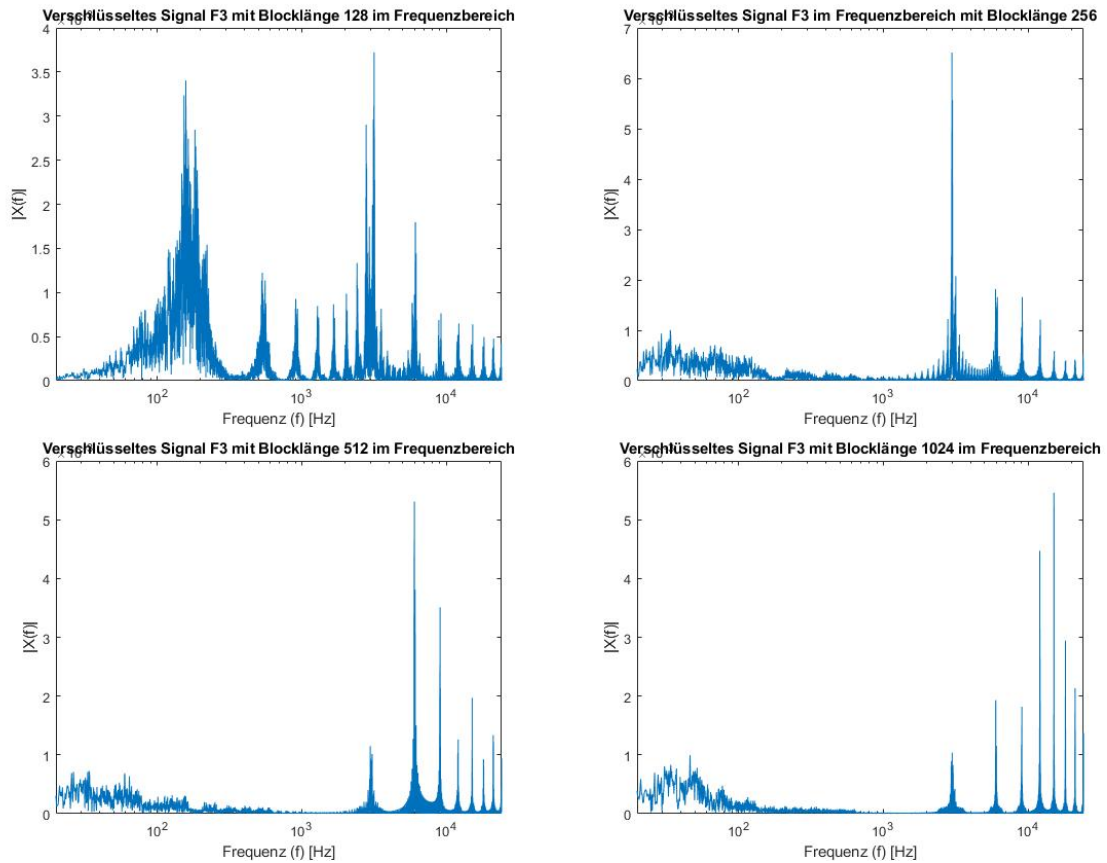
**Abbildung 2.5:** Vergleich der Spektren des Originalsignals F3 mit der Verschlüsselung des Signals mit Blocklänge 128 und Verschachtelung innerhalb des Blockes

## 2 Planung

Wie auch auf Abbildung 2.3 zu sehen, gibt es einen Peak bei etwa 158 Hz. Es gibt nun allerdings noch eine Frequenzspitze bei etwa 3100 Hz, die pegelmäßig noch über diesem liegt. Zusätzlich sind viele kleine Spitzen im Signal enthalten, die der Struktur des Originalsignals ähneln, aber ebenfalls bei unterschiedlichen Frequenzen liegen.

Die Verständlichkeit verringert sich noch weiter, wenn die Blocklänge verdoppelt wird und anschließend in 16 Blöcken zu je 16 Werten auf dieselbe Weise umstrukturiert wird. Auch eine Blocklänge von 512 mit einer Verschachtelung von 32 Blöcken mit je 16 Werten oder eine Blocklänge von 1024 Werten, die in 32 Blöcke mit je 32 Werten unterteilt werden ist möglich. Wichtig bei der Verschachtelung ist allerdings, dass die Matrizen möglichst quadratisch sein sollten. So ist sichergestellt, dass möglichst keine benachbarten Abtastwerte miteinander vertauscht werden, sondern diese möglichst weit voneinander entfernt sind. In Abbildung 2.6 sind die Spektren dieser Verschlüsselungsstrukturen für die Blocklängen 128, 256, 512 und 1024 gegenübergestellt. Auf den zugehörigen Audiodateien sind F3-128-Matlab, F3-256-Matlab, F3-512-Matlab und F3-1024-Matlab ist dies auch zu hören. Je länger der Block gewählt wird, desto mehr hochfrequente Signalanteile sind zu hören.

## 2 Planung

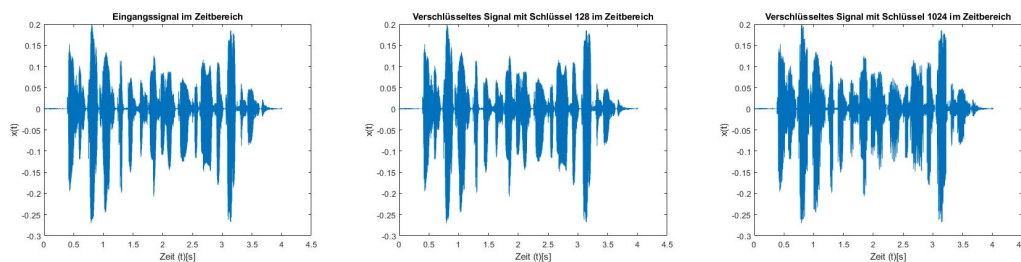


**Abbildung 2.6:** Vergleich der Spektren des mit unterschiedlichen Blocklängen verschlüsselten Signals F3-Original

Die Struktur der Sprache lässt sich allerdings auch bei einer Blocklänge von 1024 und einer Umstrukturierung von 32 Blöcken zu je 32 Werten noch erkennen. Man kann also hören, dass es einzelne Worte gibt und kann die Satzstruktur zumindest erahnen. Inhaltlich verstehen lässt sich das Signal so zwar nicht, dennoch könnte ein Hacker, dem das verschlüsselte Signal vorliegt, relativ schnell richtig schlussfolgern, dass es sich um ein Sprachsignal handelt.

Dies wird deutlich sichtbar, wenn man das unverschlüsselte Signal im Zeitbereich den verschlüsselten Signalen gegenüberstellt. Es sind hier kaum Unterschiede zu erkennen. Dies ist ein großer Nachteil, wenn es darum geht, eine hohe kryptographische Sicherheit (siehe 2.2.1) zu erreichen. Die noch klar erkennbare Wort- und Satzstruktur im Zeitbereich macht es potentiellen Mithörern vergleichsweise einfach, das Wissen über die Sprache zu nutzen, um herauszufinden, was gesprochen wurde.

## 2 Planung



**Abbildung 2.7:** Vergleich des mit unterschiedlichen Blocklängen verschlüsselten Signals F3 im Zeitbereich. Links: Original, Mitte: Schlüssel 128, Rechts: Schlüssel 1024

Um größere Unterschiede der verschlüsselten Sprachstruktur im Zeitbereich im Vergleich zum Originalsignal im Zeitbereich zu erreichen, wäre beispielsweise denkbar, die Werte des verschlüsselten Signals abwechselnd mit einem pseudozufälligen Rauschen, zum Beispiel MLS-Rauschen,<sup>1</sup> zu senden. Durch die Pseudozufälligkeit könnte dieses Signal am Empfänger erkannt und entfernt werden.

Die Blocklänge der getauschten Werte wird als Schlüssel festgelegt. So lässt sich einstellen, ob für eine sicherere Verschlüsselung im Sinne der höheren Unverständlichkeit bei einmaligem Hören eine höhere Latenz (Kapitel 2.1) in Kauf genommen wird.

### 2.2.4 Erwartete Latenzen für unterschiedliche Blocklängen

Die Latenz ist abhängig von der Blocklänge, also auch vom gewählten Schlüssel. Das liegt daran, dass für die Verschlüsselung zunächst alle Werte des Blockes abgetastet und gespeichert werden müssen und erst nach der Verschlüsselung wieder nacheinander digital-analog gewandelt werden können. Die Latenz<sup>2</sup> entspricht also der Dauer der A/D- und der D/A-Wandlung für einen Block:

$$\tau = 2n_{Block} \frac{1}{f_s} \quad [\text{s}] \quad (2.1)$$

Dies entspricht allerdings nur dem Wert der Latenz des Senders. Am Empfänger findet ebenfalls eine A/D- und eine D/A-Wandlung statt und es muss ebenfalls der gesamte Block gewandelt und gespeichert sein, bevor er entschlüsselt werden kann. Um eine ausreichende Verschlüsselung in Hinblick auf die Sprachunverständlichkeit des verschlüsselten Signals zu erreichen, ist eine Mindestblocklänge von 128 Abtastwerten erforderlich (siehe Kapitel 2.2.3).

Die Tabelle 2.1 zeigt eine Übersicht über die sich ergebenden Latenzen bei unterschiedlichen Blocklängen, berechnet mit der Abtastrate  $f_s = 30000 \text{ Hz}$ .<sup>3</sup> Zur Gesamt-

<sup>1</sup>MLS (Maximum Length Sequence) weist bestimmte Eigenschaften von Rauschen auf, ist aber dennoch deterministisch (Müller 2008: 1103)

<sup>2</sup>Die interne Verarbeitungszeit der Verschlüsselung ist hier noch nicht berücksichtigt

<sup>3</sup>Die Wahl der Abtastrate wird in Kapitel 2.3.1 erklärt.

<b>Latenzen bei unterschiedlichen Blocklängen</b>		
Blocklänge	Latenz Sender/Empfänger einzeln	Gesamtlatenz
128	8.53 ms	17.06 ms
256	17.06 ms	34.13 ms
512	34.13 ms	68.27 ms
1024	68.27 ms	136.53 ms

**Tabelle 2.1:** Latenzen bei unterschiedlichen Scrambling-Blocklängen und der Abtastrate  $f_s = 30000$  Hz

latenz addieren sich zusätzlich noch die Zeiten für die vier Wandlungen und die Zeiten für die Ver- und Entschlüsselung. Auch die Synchronisierung hat einen Einfluss auf die Latenz, darauf wird jedoch erst in Kapitel 2.3 eingegangen.

## 2.3 Synchronisierung von Sender und Empfänger

Für beide denkbaren Verfahren muss auf der Empfängerseite klar sein, wo ein einzelner Block beginnt. Es muss also zu Beginn jedes Rahmens eine Bitfolge gesendet werden, die am Empfänger möglichst auch bei einem schlechten Übertragungskanal erkannt werden kann.

### 2.3.1 Barker-Codesequenzen

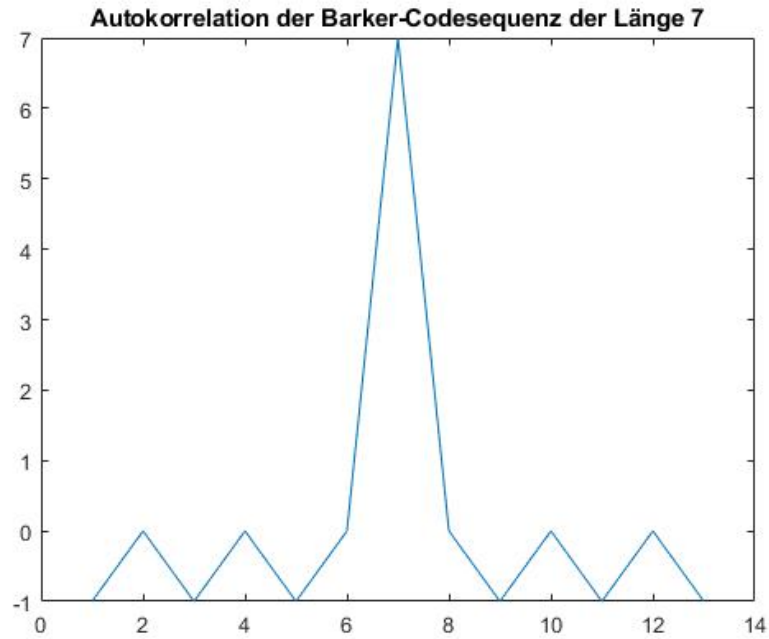
Hierzu eignen sich die unter Anderem sogenannten Barker-Codefolgen. Es gibt Barker-Codefolgen unterschiedlicher Länge, wobei 2 die kürzeste und 13 die längste Barker-Codesequenz ist. Diese haben die Eigenschaft einer geringen Autokorrelation ([Terr & Weissstein 2019](#))

Je länger die Barker-Codesequenz ist, desto höher ist ihr Maximum im Vergleich zu den Nebenkeulen. Allerdings müssen eben auch mehr Werte zusätzlich gesendet werden. Dies führt wiederum zu einer längeren Zeit am Dechiffrierer, der diese Werte abwarten muss, ehe er einen Rahmenbeginn erkennen kann. Als Kompromiss soll für die geplante Anwendung daher eine Barker-Codesequenz der Länge 7 verwendet werden, deren Autokorrelationsergebnis in Abbildung 2.8 zu sehen ist.

### 2.3.2 Unterschiedliche Abtastraten

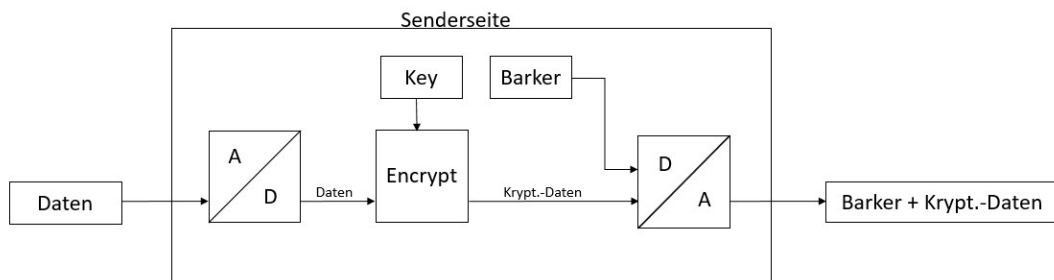
Berücksichtigt werden muss beim Hinzufügen der Barker-Codesequenz, dass hierdurch in derselben Zeit, in der eine bestimmte Anzahl an Eingangswerten abgetastet wird, mehr Werte wieder ausgegeben werden müssen. Damit also das gesamte Audiosignal gesendet werden kann, müssen die Abtastraten des Senders und des Empfängers





**Abbildung 2.8:** Autokorrelation der Barker-Codesequenz der Länge 7

am Ausgang und Eingang jeweils unterschiedlich groß gewählt werden. Die nötige Differenz hängt davon ab, wie häufig die Barker-Codesequenz gesendet werden soll. Würde dies nicht beachtet werden, könnten in der Zeit, in der die Barker-Codesequenz gesendet wird, keine Werte des Signals gesendet werden. Diese würden aber trotzdem abgetastet und würden zwangsläufig verloren gehen oder die Latenz des Systems würde sich mit jedem Senden der Barker-Codesequenz weiter erhöhen (unter Annahme eines unendlich großen Speichers).

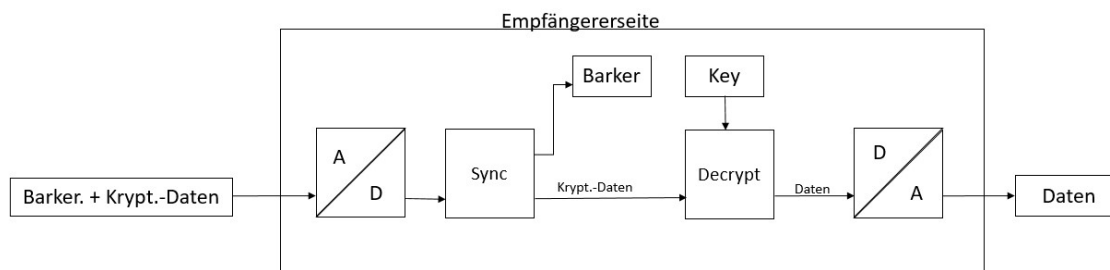


**Abbildung 2.9:** Systemstruktur der Senderseite

Auf Empfängerseite wäre dieses Problem genau gegenteilig. Die Barker-Codesequenz wird zwar am Eingang analog-digital gewandelt, wird aber nicht mehr auf den Aus-

## 2 Planung

gang geschickt. Bei gleicher Abtastfrequenz würden so zwangsläufig in bestimmten Abständen keine Daten zum Wandeln am Ausgang zur Verfügung stehen. Auch hier hängt die nötige Differenz der Abtastraten von der Häufigkeit der Barker-Codesequenz ab.



**Abbildung 2.10:** Systemstruktur der Empfängerseite

Da es zum Decodieren wichtig ist, den Beginn eines verschlüsselten Blocks zu kennen, wird die Codesequenz vor jedem Block gesendet. Die Blocklänge entspricht dem Schlüssel, somit ist auch der zeitliche Abstand zweier Barker-Codesequenzen vom gewählten Schlüssel abhängig. Eine logische Konsequenz daraus ist, dass der Unterschied der Abtastraten am Eingang und Ausgang ebenfalls in Abhängigkeit des Schlüssels variieren muss. Dies gilt jeweils für den Sender und den Empfänger, jedoch genau gegenteilig (siehe Abbildungen 2.9 und 2.10).

### **Berechnungen verschiedener Abtastratendifferenzen für verschiedene Schlüssel**

Bei einer Blocklänge von 128 müssten also auf Senderseite in derselben Zeit 128 Werte abgetastet und 135 Werte ausgegeben werden. Für eine einfachere Programmierung wird die Barker-Codefolge um einen Wert erweitert, sodass sie nun 8 Werte lang ist. Der Faktor, um den beide Abtastraten unterschiedlich sein müssen, liegt bei  $\frac{128}{136} = \frac{17}{16}$ . Nach 16 Werten dürfte also ein Eingangswert nicht abgetastet werden, während trotzdem ein Wert ausgegeben wird. Dies resultiert in einer tatsächlichen Eingangsabtastrate von  $\frac{16}{17} \cdot 30000 \text{ Hz} = 28235 \text{ Hz}$ , sodass die Eingangsnyquistfrequenz bei 14117 Hz liegt. Diese Grenzfrequenz ist jedoch für ein Sprachsignal zu vertreten, da dies ohnehin in diesem Bereich kaum Frequenzen hat.

In Tabelle 2.2 sind nun die tatsächlichen Eingangsabtastraten für verschiedene Blocklängen dargestellt.

---

**Variierende Abtastraten bei unterschiedlichen Blocklängen**


---

Blocklänge	Abtastfrequenz Eingang	Abtastfrequenz Ausgang
128	28235 kHz	30 kHz
256	29091 kHz	30 kHz
512	29538 kHz	30 kHz
1024	29767 kHz	30 kHz

---

**Tabelle 2.2:** Vergleich unterschiedlicher Abtastraten für verschiedene Blocklängen des Chiffrierers

Der Unterschied beider Abtastraten wird also kleiner, je länger der Block gewählt wird, da die Barker-Codesequenz jeweils nur am Anfang eines Blockes gesendet wird. Auf der Empfängerseite sind dies die Ausgangsabtastraten, sodass das System von außerhalb betrachtet mit der jeweils geringeren Abtastfrequenz arbeitet. Für die Übertragung innerhalb des Systems über die Mikrofonfunkstrecke beträgt die Abtastrate allerdings immer 30 kHz.

## 2.4 Kanaleinfluss

Alle nötigen Übertragungswege, die das verschlüsselte Signal im geplanten Systemaufbau (Abbildung 1.1) hat, erfolgen über einen analogen Kanal. Physikalisch werden als Übertragungsmedium Kabel verwendet sowie ein Funkkanal, der durch die Mikrofonfunkstrecke bereitgestellt wird. Doch unabhängig vom Übertragungsmedium muss bei der Übertragung durch einen analogen Kanal mit Störeinflüssen gerechnet werden.

Jeder analoge Kanal besitzt eine obere und eine untere Grenzfrequenz, die die Bandbreite des Durchlassbereichs definieren. Zusätzlich wird das Signal durch weitere Effekte wie Rauschen und Störsignale beeinflusst, die sich bei der Kanalübertragung dem Signal hinzuaddieren. Dazu zählen beispielsweise Widerstands- und Halbleiter-rauschen, Quantisierungsrauschen, Übersprechen, Metzeinstreuungen sowie Einstreuungen durch Schaltvorgänge. Außerdem können bei der Übertragung lineare oder nichtlineare Verzerrungen entstehen ((Meyer 2014: 38)).

Da die meisten Kanäle linear sind, können sie als LTI-System<sup>4</sup> betrachtet werden ((Meyer 2014: 352)) und folgendermaßen beschrieben werden können:

$$y(t) = x(t) * h(t) + n(t) \quad (2.2)$$

Interessant ist nun, ob das verschlüsselte Signal diesen Einwirkungen des Kanals standhält.

---

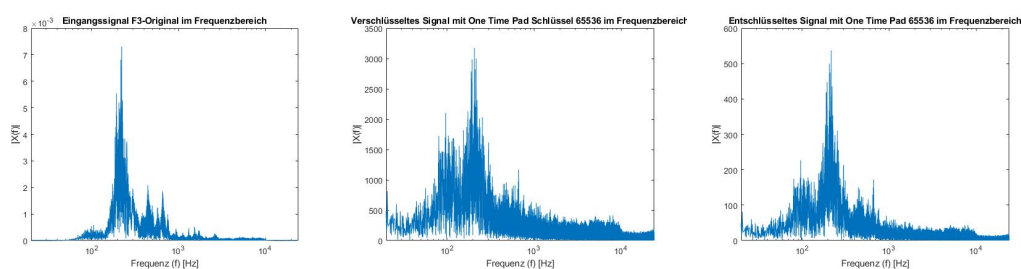
<sup>4</sup>Linear Time Invariant

### 2.4.1 One Time Pad über Kanal

Betrachtet man den geplanten Systemaufbau 1.1, würde unter Annahme eines idealen Kanals mit idealen Systemkomponenten eine bestimmte Spannung am Mikrofon erzeugt und anschließend verstärkt werden. Diese würde dann am Eingang des Chiffrierers einen bestimmten Abtastwert  $X$  liefern. Durch die XOR-Operation mit dem Schlüssel wird daraus Wert  $Y$ . Auch dieser Wert führt zu einer exakten, aber anderen Spannung, die über die Mikrofonfunkstrecke übertragen wird und auf Empfängerseite am Eingang des Dechiffrierers vom A/D-Wandler wieder zu Wert  $Y$  abgetastet wird. Nach der Entschlüsselung durch eine erneute XOR-Operation ergibt sich daraus wieder der Wert  $X$ .

Da die Übertragung aber nicht nur in der Theorie über einen idealen Kanal, sondern auch in der Praxis über einen realen Kanal funktionieren soll, muss das Verschlüsselungsverfahren resistent gegenüber den Einflüssen eines realen Kanals sein (Kapitel 2.4).

Schon eine geringe Kanaldämpfung kann je nach Spannungsunterschied pro Quantisierungsstufe des Wandlers zu einem anderen Abtastwert führen. Auch ein Rauschen, was sich bei der Kanalübertragung zum Signal hinzuaddiert, könnte dazu führen. Somit könnte der zuvor gesendete Wert  $Y$  am A/D-Wandler des Empfängers nicht mehr zum Wert  $Y$  abgetastet und quantisiert werden. So ergibt sich bei der nachfolgenden Dechiffrierung auch nicht mehr der Originalwert  $X$ .

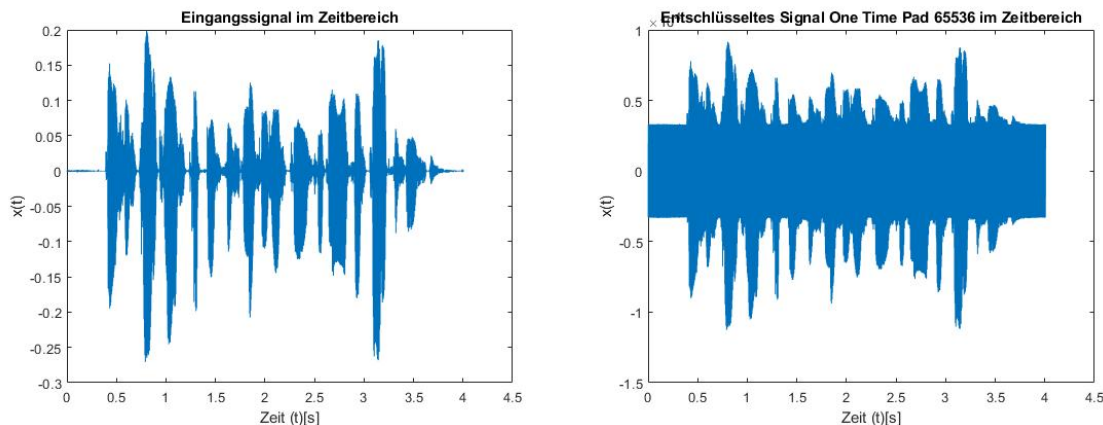


**Abbildung 2.11:** Vergleich der Spektren des Originalsignals F3 mit dem verschlüsselten und entschlüsselten Signal (OTP) nach Kanalübertragung

Es ist deutlich zu erkennen, dass das Spektrum des entschlüsselten Signals dem des verschlüsselten Signals mehr ähnelt als dem originalen. Das Signal wurde hierfür über einen simulierten Kanal mit einer Dämpfung um den Faktor 0,9 und einem SNR<sup>5</sup> von 80 dB ohne Verzögerung übertragen. Betrachtet man das originale und das entschlüsselte Signal im Zeitbereich, wird der Unterschied aber noch deutlicher.

---

<sup>5</sup>SNR=Signalrauschabstand



**Abbildung 2.12:** Vergleich des Originalsignals F3 mit dem entschlüsselten Signal (OTP) nach Kanalübertragung im Zeitbereich

Auch das entschlüsselte Signal ist ohne Probleme zu verstehen (Audiofile F3-OTP65536-K), enthält aber, wie auch auf Abbildung 2.12 gut zu sehen ist, weiterhin Störanteile. Digitale Systeme, deren Verschlüsselungsverfahren auch Substitutionen enthalten lösen dieses Problem durch eine Kanalcodierung auf Senderseite. So können auftretende Bitfehler am Empfänger erkannt und wieder korrigiert werden.

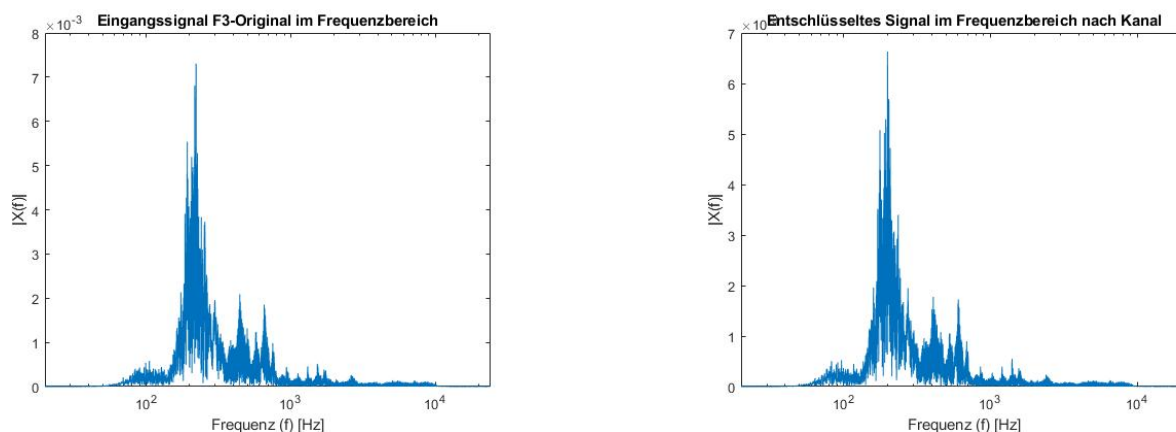
Da eine Kanalcodierung im Rahmen dieser Arbeit nicht geplant ist, sind Substitutionsverfahren, zu denen auch das One Time Pad gehören, für den angestrebten Anwendungsfall also nicht zweckmäßig.

### 2.4.2 Scrambling über Kanal

Wenn das Scrambling sich nicht auf das Vertauschen einzelner Abtastwerte, sondern auf die Bits, aus denen sich jeder einzelne Abtastwert zusammensetzt, besteht, gäbe es bei der Übertragung über einen realen Kanal ein ähnliches Problem wie beim One Time Pad. Wenn sich die Spannung zum Beispiel durch eine Dämpfung so ändert, dass dies zu einem anderen Abtastwert am nächsten Wandler führt, könnte das Originalsignal nicht mehr zurückgewonnen werden. Nimmt man an, dass das Signal mit 16 Bit abgetastet wurde und diese Bits sich in ihrer Reihenfolge vertauschen, wird das LSB zum MSB<sup>6</sup>. Bei der Kanalübertragung ist es wahrscheinlicher, dass sich das LSB nach der nächsten Abtastung verändert hat, als das MSB. Beide vertauschen sich jedoch bei der Entschlüsselung wieder. Da das MSB einen höheren Einfluss darauf hat, welche Spannung den 16 Bits entsprechen, sorgt die Änderung dieses Bits für einen völlig anderen Wert. Daher wäre auch hier eine Kanalcodierung nötig. Anders verhält es sich, wenn nur komplette Abtastwerte beim Scrambling vertauscht werden. Wird das analoge Signal nun bei der Übertragung geringfügig verändert, sodass etwas

<sup>6</sup>LSB = Least Significant Bit, MSB = Most Significant Bit

andere Abtastwerte nach der Wandlung vorliegen, werden nun diese etwas anderen Abtastwerte miteinander wieder vertauscht, aber der einzelne Abtastwert verändert sich nicht. Daher lassen sich nach der Übertragung des Signals über einen Kanal mit einer Dämpfung um den Faktor 0,9 und einem SNR von 80 dB auch keine Unterschiede in den Spektren und den Audiodateien (F3-128-Matlab und F3-128-Matlab-K1) erkennen.



**Abbildung 2.13:** Links: Spektrum des verschlüsselten Signals (128), Rechts: Spektrum des Signals nach Senden über Kanal

### 2.4.3 Kanaleinfluss auf Barker-Codesequenz

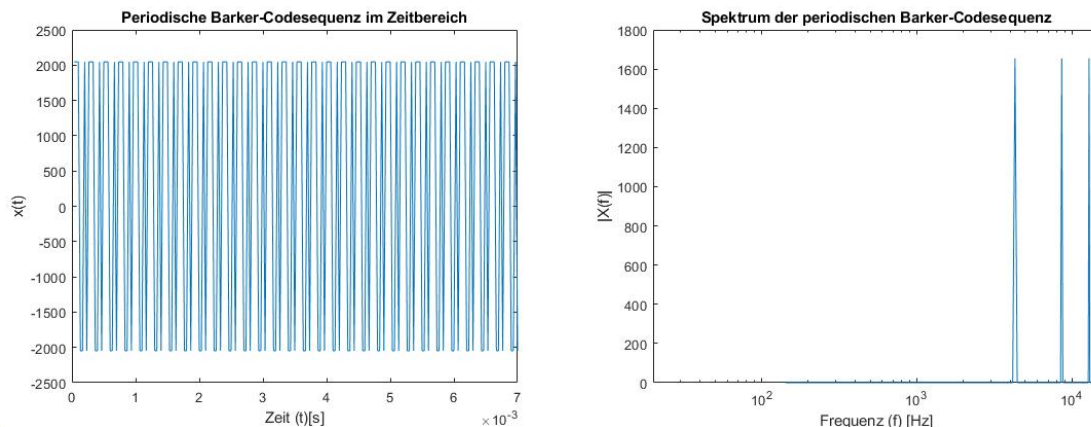
Eine Signalübertragung über einen Kanal ist nur möglich, wenn der Frequenzbereich des zu sendenden Signals zur Übertragungsbandbreite des Kanals passt (Meyer 2014: 39). Die obere Grenzfrequenz der PGX-Funkstrecke liegt bei 15kHz (Shure 2010). Daher darf auch das Spektrum der Barker-Codesequenz keine höheren Frequenzen aufweisen. Die Frequenzen des Spektrums hängen allerdings maßgeblich von der gewählten Abtastrate ab. Die höchstmögliche Frequenz wird durch das Nyquist-Theorem  $f_{abast} > 2 \cdot f_{signal}$  bestimmt. Da in diesem Fall aber das Signal digital erzeugt wird, entspricht die höchste maximal erreichbare Frequenz einem periodischen Wechsel zweier Abtastwerte, deren Periodendauer genau zwei Werten entspricht. Die daraus resultierende Frequenz liegt genau bei der Nyquist-Frequenz <sup>7</sup>.

Die Barker-Codesequenz beinhaltet solche Wechsel, die zu dieser theoretisch möglichen Höchstfrequenz führen. Betrachtet man das Spektrum, lässt sich feststellen, dass es, wie zu erwarten war, sehr hohe Frequenzanteile enthält. Die Höchstfrequenz liegt bei 12860 Hz, zwei weitere Frequenzen bei 8571 Hz und 4286 Hz.

---

<sup>7</sup>  $f_{nyquist} = \frac{1}{2} f_{abast}$

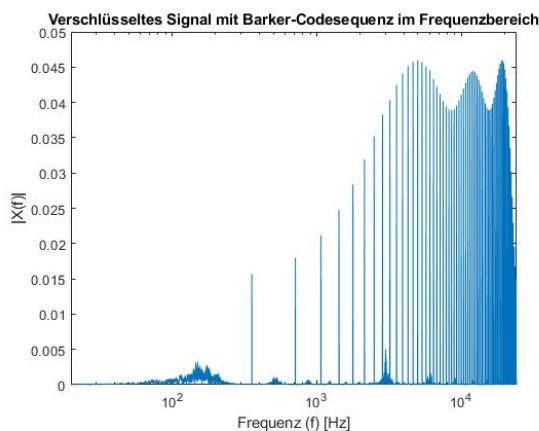
## 2 Planung



**Abbildung 2.14:** Periodische Barker-Codesequenz im Zeit- und Frequenzbereich

Damit sichergestellt ist, dass durch die Barker-Codesequenz keine Frequenzen entstehen, die über den analogen Kanal nicht übertragen werden können, wird eine Abtastfrequenz in Höhe von 30 kHz gewählt.

### 2.4.4 Gesamtsignal über Kanal

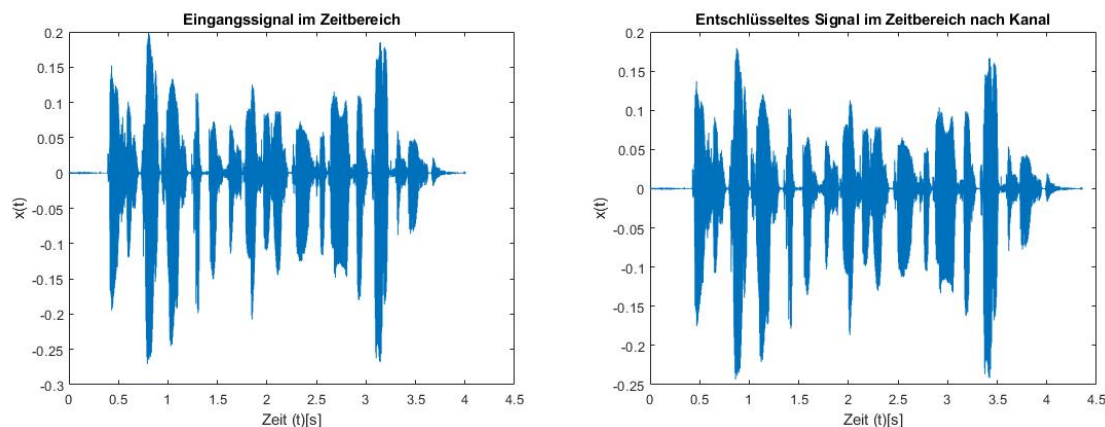


**Abbildung 2.15:** Spektrum des verschlüsselten Signals mit Schlüssel 128 und der Barker-Codesequenz

Das Spektrum des Gesamtsignals, das aus dem Spektrum der Barker-Codesequenz und dem des verschlüsselten Signals besteht, wird nun über den Kanal mit einer Dämpfung von 0,9 und einem SNR von 80dB gesendet. Nach der Übertragung lässt sich das entschlüsselte Signal wieder ohne Probleme verstehen, das Rauschen ist kaum wahrnehmbar (Audiofile F3-Matlab-128-K1). Wird das entschlüsselte Signal mit dem

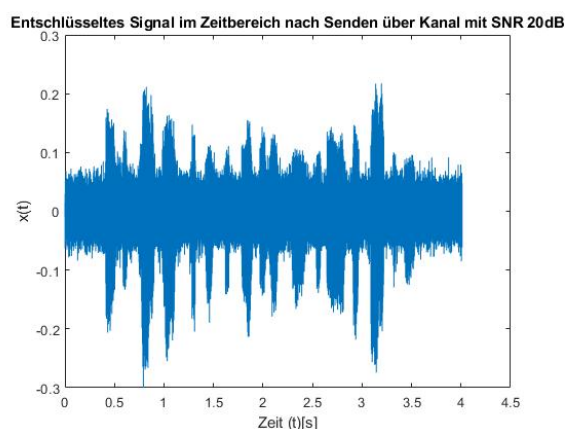
## 2 Planung

Originalsignal im Zeitbereich verglichen, ist ebenfalls kaum ein Unterschied erkennbar.



**Abbildung 2.16:** Darstellung der Verschlüsselungsstruktur bei einer Blocklänge von 128 Werten

Im Vergleich zur Nutzung des One Time Pads ist hier kaum ein Unterschied zu erkennen. Erst, wenn der Signalrauschabstand deutlich geringer wird, ist das Rauschen zu hören (Audiofile F3-Matlab-128-K2) und bei der Darstellung im Zeit- und Frequenzbereich zu sehen. Hier beträgt der Signalrauschabstand des Kanals 20 dB. Wird der Klangeindruck dieses Signals mit dem entschlüsselten Signal, das mittels One Time Pad verschlüsselt wurde (F3-OTP65536-K), ist dies ein deutlich anderer. Durch die fehlerhafte Dechiffrierung beim One Time Pad treten deutlich hörbare Verzerrungen auf, die in diesem Fall nicht vorhanden sind.



**Abbildung 2.17:** Spektrum des verschlüsselten Signals mit Schlüssel 128 und der Barker-Codesequenz



## 2.5 Signale und Pegel an Übergangspunkten

### 2.5.1 Chiffrierer

#### Eingang

Wie in Abbildung 1.1 zu sehen ist, wird soll das Mikrofon direkt an die Verschlüsselungsendeeinheit angeschlossen werden. Die Angabe des Feldübertragungsfaktors  $B_e$  beziehungsweise der Empfindlichkeit in Datenblättern von Mikrofonen gibt Aufschluss darüber, in welche elektrische Spannung das Mikrofon einen eintreffenden Schalldruckpegel wandelt.

$$B_e = \frac{\tilde{u}}{\tilde{p}} \quad \left[ \frac{\text{V}}{\text{Pa}} \right] \quad (2.3)$$

Für das später in den Messungen benutzte SM58 von Shure, beträgt dieser Wert 1,85 mV ((Shure 2014:6)). Der Pegel von normaler Sprache liegt bei etwa 60dB (Umweltbundesamt 2014). So lässt sich nun berechnen, wie viel Spannung am Ausgang des Mikrofons anliegt.

$$\tilde{p} = 10^{\frac{L}{20}} \tilde{p}_0 \quad \left[ \frac{\text{V}}{\text{Pa}} \right] \quad (2.4)$$

Mit  $p_0 = 2 \cdot 10^{-5}$  [Pa] ergibt sich so ein Schalldruck von 0,02 Pa. Löst man nun die Formel 2.3 nach  $\tilde{u}$  auf, ergibt sich ein Spannungseffektivwert in Höhe von 37  $\mu\text{V}$ . Da die Mikrofonfunkstrecke PGX von Shure ein Consumer-Gerät ist, werden die folgenden Berechnungen auch unter Berücksichtigung des anderen Spannungsreferenz- und Normpegels durchgeführt<sup>8</sup>. Die am Mikrofon anliegende Spannung entspricht -88 dBV. Damit dieser Spannungswert auf den Consumerpegel gebracht wird, ist eine Verstärkung um um 78 dB notwendig, was einem Verstärkungsfaktor von knapp 8000 entspricht.

Für die spätere Umsetzung des geplanten Systems reicht es jedoch nicht aus, den Effektivwert der Spannung zu kennen, sondern die Spitzenwerte sind ebenfalls von Interesse. Diese sind jedoch je nach Art des Eingangssignal unterschiedlich, In diesem Fall handelt es sich um Sprachsignale. Der Spitzenwert lässt sich nun aus dem Crest-Faktor berechnen. Dieser gibt das Verhältnis zwischen Spitzen- und Effektivwert in dB an.

$$C = \frac{\hat{x}}{\tilde{x}} \quad (2.5)$$

Für Sprachsignale liegt der Crest-Faktor bei etwa 12 bis 20 dB (Goertz & Schmitz 2012: 20), das entspricht einem Faktor von 3,98 bis 10. So lässt sich nun der Spitzenwert berechnen des verstärkten Mikrofonsignals berechnen. Bei der Berechnung mit 20 dB Crest-Faktor, liegt der Spitzenwert des unverstärkten Mikrofonsignals also bei

---

<sup>8</sup>Statt 0,775 V im Profibereich gelten 1 V im Consumerbereich als Referenz. Der Consumer-Normpegel liegt bei -10dBV

370  $\mu\text{V}$ , ein Signal auf Consumerpegel hätte einen Spitzenwert von 10 V. Bei einem angenommenen Crest-Faktor läge dieser Wert bei 3,98 V.

Das Verschlüsselungssystem bekommt also am Eingang Signale, deren Spitzenwerte bei Annahme eines Crest-Faktors von 12 dB von 3,98 V bis 10 V für einen Crest-Faktor von 20 dB reichen.

### Ausgang

Der Ausgang der Verschlüsselungsendeeinheit wird an den Taschensender angeschlossen, der die anschließende Funkübertragung durchführt. Taschensender sind so gebaut, dass daran ein Mikrofon angeschlossen werden kann, sie enthalten also einen Mikrofonvorverstärker. Für die Übertragungssicherheit wäre es aber besser, das Signal möglichst nicht wieder in den Millivoltbereich bringen zu müssen. Da der Signalrauschabstand kleiner wird, je geringer das Signal angesteuert wird, wäre es von Vorteil, wenn das verschlüsselte Signal über die gesamte Übertragungstrecke einen ausreichend hohen Pegel hätte. Der Taschensender PGX1 von Shure besitzt allerdings drei Stellmöglichkeiten und lässt so auch höhere Pegel als Eingangssignal zu (Shure 2010). Daher kann das Signal von der Verschlüsselungsendeeinheit mit Consumer-Pegel ausgegeben werden, wenn der Taschensender auf die Position „0“ gestellt wird. Da die dritte Position das Signal sogar um 10dB absenkt, könnte sogar der Funkhausnormpegel verwendet werden, ohne das System zu übersteuern.

### 2.5.2 Dechiffrierer

Der verwendete Funkempfänger besitzt einen symmetrischen und einen unsymmetrischen Ausgang, wobei der Ausgangspegel für den symmetrischen Ausgang bei -19 dBV und beim unsymmetrischen bei -5 dBV liegt (Shure 2010). Es soll der Klinkenausgang genutzt werden, an den der Dechiffrierer angeschlossen wird. Dieser muss also genauso wie der Chiffrierer in der Lage sein, Signale, die mit Consumer-Pegel angesteuert sind, zu verarbeiten. An die Entschlüsselungseinheit soll nun, wie sonst an den Funkempfänger, ein Lautsprecher oder ein Mischpult angeschlossen werden können. Hierfür wird also ebenfalls der Consumer-Pegel gewählt, da auch der PGX4 diesen Pegel an seinen Ausgängen bereitstellt.

## 2.6 Mobilität und Nutzbarkeit

Damit das geplante System die übliche Nutzung einer Mikrofonfunkstrecke nicht einschränkt, muss die Verschlüsselungseinheit auf Senderseite ebenso mobil sein wie die Sendeeinheit der Mikrofonfunkstrecke. Ansonsten ginge der Sinn der drahtlosen Übertragung, nämlich die gewonnene Mobilität und der Wegfall von Kabeln, verloren. Daher soll die Verschlüsselungsendeeinheit so ausgeführt werden, dass sie grundsätzlich mobil ist. Allerdings ist es nicht das Ziel, ein praktisches Gehäuse zu entwickeln wie

## 2 Planung

es beispielsweise Taschensender durch die Befestigungsmöglichkeit an der Kleidung üblicherweise haben.

Damit die Mobilität erreicht wird, wird für die Senderseite der Betrieb mit einer Batterie geplant. Die Batterie schränkt das System in ihrer Nutzungsdauer ein. Dies ist allerdings ohnehin bei der Verwendung einer Mikrofonfunkstrecke der Fall. Es sollte lediglich vor der Nutzung kontrolliert werden, ob die verwendeten Batterien ausreichend aufgeladen beziehungsweise neu sind.

Für die Empfängerseite ist die Mobilität nicht wichtig, trotzdem sollte eine einfache Bedienung ermöglicht werden. Für einige Anwendungen wäre es zum Beispiel von Vorteil, wenn sich die Empfängereinheit in ein 19-Zoll-Rack bauen ließe. Außerdem sollte bei einem Wechsel des Schlüssels die Übertragung des neuen Schlüssels möglichst einfach erfolgen. Am Markt vorhandene Systeme lösen dies meist über die ohnehin schon vorhandene Infrarot-Schnittstelle ([Schwörer 2019](#), [Gmoser 2019](#)).

Diese Überlegungen sollen Anregungen für zukünftige Weiterentwicklungen sein, eine Umsetzung besonders praktischer Gehäuse ist nicht geplant. Lediglich die Mobilität des Chiffrierers soll erreicht werden.

# 3 Umsetzung

## 3.1 Wahl der Plattform

Als Umsetzungsplattform bietet sich ein Mikrocontroller an. Es gibt verschieden Typen von Mikrocontrollern, die verschiedene Eigenschaften haben.

Die Anforderungen an den Mikrocontroller sind die Bereitstellung eines A/D- und eines D/A-Wandlers. Außerdem muss die interne Rechengeschwindigkeit hoch genug sein, um Audiosignale in Echtzeit verarbeiten zu können. Es sollte außerdem die Möglichkeit gegeben sein, die Wandler per Interrupt-Logik zeitzusteuern. Das Auftreten eines Interrupts, der zum Beispiel durch den A/D-Wandler ausgelöst wird, unterbricht die momentane Aufgabe des Mikrocontrollers, um diesen Wert zu verarbeiten (Demowski 2014: 14). Darüber hinaus sollte eine Speichermöglichkeit vorhanden sein, da für das geplante System Abtastwerte gespeichert werden müssen.

Diese Anforderungen erfüllen zum Beispiel DSPs<sup>1</sup>, die besonders durch eine sehr schnelle Verarbeitung gekennzeichnet sind (Gessler 2014: 58). Da Mikrocontroller allerdings eine komfortablere Programmierumgebung bieten und zusätzlich bereits alle nötigen Ein- und Ausgänge bereitstellen, soll für die Umsetzung ein solcher verwendet werden. Hier eignet sich der Arduino Due, der sowohl ADCs<sup>2</sup> als auch zwei DACs<sup>3</sup> besitzt. Außerdem kann der verbaute Chip, ATMEL3X8E durch seine interne Taktfrequenz von 84 MHz die Audiosignale ausreichend schnell verarbeiten (Atmel 2015). Zudem bieten die Wandler eine Bittiefe von 12 Bit an, was für Audiosignale zwar geringer ist, als CD-Qualität<sup>4</sup>, aber zunächst ausreichend erscheint.

## 3.2 Analoge Schaltungen

Damit das System zu einer Mikrofonfunkstrecke kompatibel ist, müssen durch entsprechende analoge Schaltungen die schon in Kapitel 2.5 berechneten Spannungswerte und Pegel an den jeweiligen Ausgängen vorliegen, beziehungsweise an den Eingängen eingespeist werden können. Der Eingang des A/D-Wandlers des Arduino Due kann Spannungswerte von 0 V bis +3,3 V verarbeiten. Am Ausgang können Spannungswerte im Bereich von 0,55 V bis 2,75 V ausgegeben werden (Atmel 2015).

---

<sup>1</sup>Digital Signal Processor

<sup>2</sup>Analog Digital Converter, deutsch: A/D-Wandler

<sup>3</sup>Digital Analog Converter, deutsch: D/A-Wandler

<sup>4</sup>CDs sind mit 16 Bit quantisiert

Es werden also prinzipiell drei Schaltungen benötigt, die folgende Aufgaben erfüllen müssen:

1. Mic-In zu ADC In
2. DAC Out zu Consumer-Pegel
3. Consumer-Pegel zu ADC In

Die dritte Schaltung kann für die Sende- und Empfangsseite grundsätzlich identisch sein, da die Schaltung die selbe Aufgabe erfüllt, den Spannungsbereich des DAC jeweils auf Consumer-Pegel zu bringen. Auf die umgesetzten Unterschiede wird in den Kapiteln 3.2.1 und 3.2.2 genauer eingegangen.

#### 3.2.1 Chiffrierer

(Carrera 2015) liefert bereits einen Vorschlag für eine Schaltung, die ein bipolares Signal so für den Arduino Due aufbereitet, dass dieser es verarbeiten kann. Als zulässige Eingangsspannungen sind in dieser Schaltung Werte von  $-3,3\text{V}$  bis  $+3,3\text{V}$  angegeben. Dieselben Werte sind auch die Minimal- und Maximalwerte, die am Ausgang anliegen können. Da an den Sender allerdings nur ein Mikrofon angeschlossen wird, dieser Schaltung eine Mikrofonvorverstärkerschaltung vorgeschaltet (Conrad Electronic GmbH 1999). Sie wird wie im Datenblatt angegeben so umgesetzt, dass sie für dynamische Mikrofone funktioniert. Die Schaltung besitzt eine 150- bis 1500-fache Verstärkung. Das Mikrofonsignal, das zuvor in Kapitel 3.1 berechnet wurde, besitzt daher nach der Vorverstärkung einen Spannungseffektivwert von  $55,5\text{ mV}$  und einen Spitzenwert von  $555\text{ mV}$ . Da dieser Wert deutlich unter den Maximalwerten der Schaltung von (Carrera 2015) liegt, wird in dieser ein zusätzlicher Kondensator über R3 parallel geschaltet. Dieser sorgt dafür, dass das Eingangssignal lediglich einen DC-Offset erhält, aber nicht bedämpft wird. Allerdings wird so trotzdem nicht der geplante Pegel erreicht. Deshalb wird hinter den Mikrofonvorverstärker eine zusätzliche Klinkenbuchse integriert, die es ermöglicht, einen externen Mikrofonvorverstärker zu nutzen. Außerdem können mit einem externen Vorverstärker auch Kondensatormikrofone genutzt werden, was mit der umgesetzten Schaltung von (Conrad Electronic GmbH 1999) nicht möglich ist. Das Tiefpassfilter am Eingang der Schaltung (Carrera 2015) besitzt eine Grenzfrequenz von  $220\text{ Hz}$ , die für die geplanten Zwecke zu niedrig ist. Die Bauteilwerte werden daher so gewählt, dass eine höhere Grenzfrequenz erreicht wird. Das Filter soll als Antialiasingfilter fungieren und muss deshalb in der Lage sein, Frequenzen oberhalb der halben Abtastfrequenz ausreichend zu bedämpfen, um Aliasing<sup>5</sup> zu verhindern. Da es sich nur um ein Filter 2. Ordnung handelt, ist das Ausgangssignal an der Grenzfrequenz des Filters nur um  $-6\text{ dB}$  bedämpft. Daher wird die Grenzfrequenz schon auf  $10\text{ kHz}$  festgelegt, sodass die Dämpfung bei  $15\text{ kHz}$

---

<sup>5</sup>Aliasing sind Frequenzen, die durch eine Unterabtastung entstehen, wenn das Nyquisttheorem nicht erfüllt wird

### 3 Umsetzung

bereits etwa 12 dB beträgt. Diese Umsetzung ist nicht ideal, da so auch Frequenzen, die eigentlich bei vollem Pegel übertragen werden sollten, bedämpft werden. Um das System zu verbessern, sollte also ein Filter höherer Ordnung mit einer noch höheren Grenzfrequenz verbaut werden.

Der ICL7660 aus der Schaltung von (Carrera 2015) dient dazu, einen der beiden LM358 mit einer negativen Spannung zu versorgen, damit dieser auch Spannungen unterhalb von 0 V ausgeben kann. Allerdings stört der ICL7660 das Signal, indem er mit einer Frequenz in Höhe von etwa 5 kHz mitschwingt und wird daher entfernt. Die negative Spannungsversorgung übernimmt stattdessen eine Batterie. Dies ist nicht ideal, da die zusätzliche Batterie die Mobilität einschränkt, also wäre auch hier noch Potenzial für Verbesserungen.

Eine weitere Modifikation der Schaltung ist die Anpassung des Kondensators C5 (A.3). Mit einem Wert von 100 nF, wie er in der Schaltung von (Carrera 2015) zu finden ist, besitzt er eine Grenzfrequenz von 159 Hz und verhindert somit die Verstärkung. Da er jedoch dazu dient, Störsignale zu entkoppeln, wird ein Kompromiss gewählt, sodass ein Kondensator mit 22 nF verbaut wird, sodass die Grenzfrequenz bei 723 Hz liegt.

$$f_g = \frac{1}{2\pi\sqrt{RC}} \quad [\text{Hz}] \quad (3.1)$$

Ebenfalls zur Störsignalentkopplung wird ein zusätzlicher Kondensator mit 1 nF zwischen den Pins 2 und 6 des Operationsverstärkers LF351 in der Mikrofonvorverstärkerschaltung hinzugefügt. Auch der Mikrofoneingang wird durch einen Kondensator in Höhe von 100 nF, der nach Masse geschaltet ist, entkoppelt.

Da der Operationsverstärker LM358N nicht kapazitiv belastet werden kann, ohne, dass es zu Signalstörungen kommt, wird ein Ausgangswiderstand in Höhe von 100  $\Omega$  nachgeschaltet.

#### 3.2.2 Dechiffrierer

Auf Empfängerseite wird die Schaltung (Carrera 2015) ebenfalls modifiziert. Das Tiefpassfilter ist nun ans Ende der Schaltung verschoben worden, um die Rechteckanteile des Signals, das der DAC ausgibt, zu glätten. Auch hier wurde ein 100  $\Omega$ -Widerstand vor den Ausgang geschaltet und ein zusätzlicher Kondensator zur Eingangssignalentkopplung in die Schaltung integriert.

Umgesetzt werden beide Schaltungen auf einem Shield, einer Platine, die der Größe des Arduino Due angepasst ist und sich daher leicht und platzsparend mit diesem verbinden lässt (Abbildung 3.1).

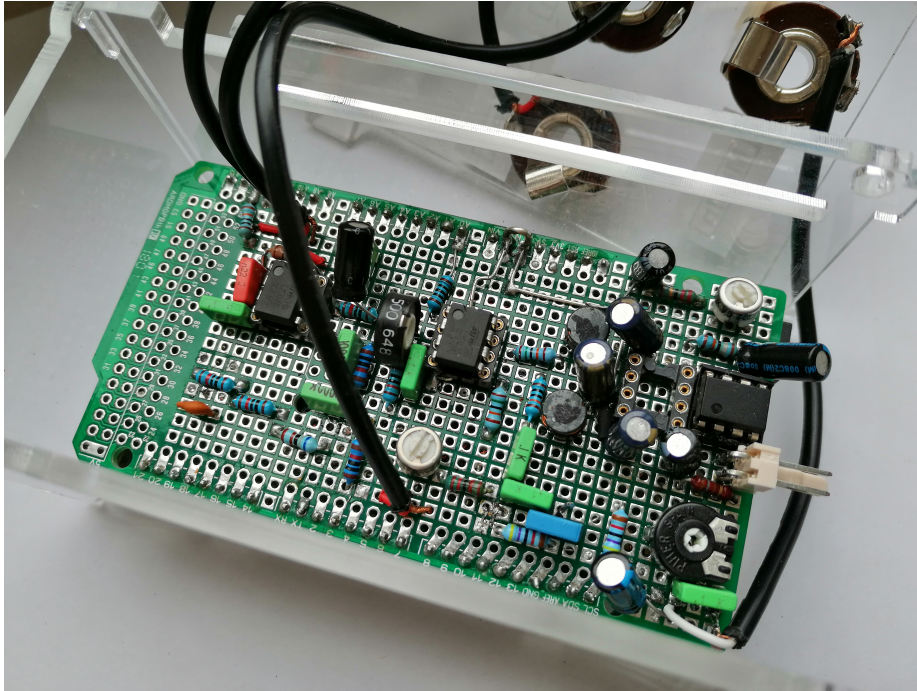


Abbildung 3.1: Foto des Chiffrierers

#### 3.2.3 Anschlüsse und Impedanzen

Damit das System insgesamt kompatibel und einfach zu verbinden ist, müssen Anschlüsse bereitgestellt werden. Alle Anschlüsse <sup>6</sup> werden als 6,3mm Klinkenbuchse ausgeführt, bis auf den DAC-Output und den ADC-Input. Diese stellen lediglich die Möglichkeit bereit, beide Teile des Verschlüsselungssystems so direkt wie möglich zu verbinden, sind aber eigentlich nicht für den Gebrauch gedacht. Diese sind daher auch aus Platzgründen im Gehäuse als 3,5mm Klinkenbuchse ausgeführt. Da die Schaltung für symmetrische Signale nicht ausgelegt ist, müssen XLR-Anschlüsse adaptiert werden. Auch für die Verbindung mit dem Taschensender ist ein Adapter nötig, da dieser einen TA4F-Anschluss besitzt. (Shure 2010)

In der Audiotechnik wird mit Spannungsanpassung gearbeitet, das heißt, dass Eingänge eine hohe und Ausgänge eine niedrige Impedanz haben sollten. So wird sichergestellt, dass die Spannung an den richtigen Stellen, nämlich den Lasten abfällt und dort für einen höheren Pegel sorgt. In diesem Fall sind vor allen Ausgängen Operationsverstärker verbaut, die idealerweise keinen Ausgangswiderstand haben. Diese bestimmen maßgeblich den Ausgangswiderstand der Schaltung, um einen genauen Wert nennen zu können, müsste dieser jedoch messtechnisch erfasst werden.<sup>7</sup>

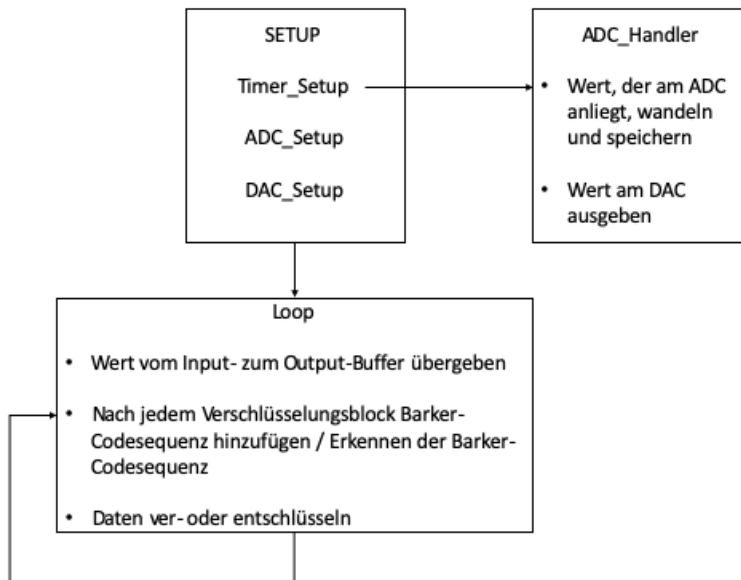
<sup>6</sup>Chiffrierer: Mic-Input, Input, Output, DAC-Output; Dechiffrierer: Input, ADC-Input, Output

<sup>7</sup>„Allgemein gilt ein Impedanzverhältnis zwischen Quelle und Empfänger von 1:5 als ausreichend (DIN EN 60268-4)“ (Schneider 2008: 376)

### 3.3 Programmierung

Im Anhang A.4 befinden sich die Programmablaufpläne für den Chiffrierer und den Dechiffrierer, mit denen sich die folgenden Kapitel besser nachvollziehen lassen.

#### 3.3.1 Programmstruktur



**Abbildung 3.2:** Schematischer Aufbau des Programms

Der Programmcode ist bei der Programmierung eines Arduinos grundsätzlich in zwei Teile aufgeteilt, den Setup-Teil und den Loop-Teil. Der Setup-Teil wird nur einmal ausgeführt, wohingegen der Loop-Teil nach jedem Durchlauf automatisch wiederholt wird. Die grundsätzliche Aufgabe, ein Signal, das am Eingang des A/D-Wandlers anliegt, mit einer bestimmten Abtastfrequenz zu wandeln und nach der Ver- oder Entschlüsselung wieder auszugeben, müssen sowohl die Sender- als auch die Empfängerseite erfüllen. Diese Programmstruktur wurde von (Newton 2015) übernommen und anschließend modifiziert.

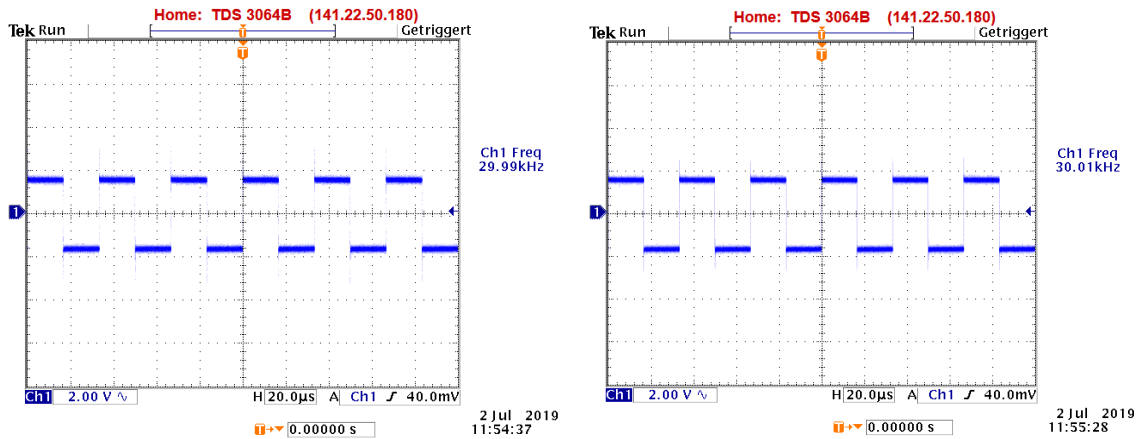
Das Programm nutzt dafür Interrupts, die je nach Einstellungen eines Timers vom ADC gesendet werden, sodass mit einer bestimmten Abtastfrequenz Werte eingelesen werden können.

Hierbei wird die interne Clock1 des Arduinochips SAM8XE genutzt, deren Frequenz halb so groß wie die interne Taktfrequenz ist und somit bei 42 MHz liegt ((Atmel 2015)). Durch Setzen der Werte kann nun die tatsächliche Frequenz des



### 3 Umsetzung

Timers gewählt werden. Da für die Übertragung über die Mikrofonfunkstrecke die maximale Abtastrate bei 30 kHz liegt, wird der TC-RC auf 1400 gesetzt. Dadurch beträgt die Frequenz des Timers nun  $\frac{42\text{MHz}}{1400} = 30\text{kHz}$ . Über die Funktion „setup-pio-TIA00“ kann die konfigurierte Clock auf Digital Pin 2 des Arduinos ausgegeben und sich so kontrollieren lassen.



**Abbildung 3.3:** Timer auf Sender- und Empfängerseite mit einer Frequenz von 30 kHz

Der ADC wird im Setup so konfiguriert, dass er durch den Timer getriggert wird. Außerdem wird Pin A0 als Eingang festgelegt. Im DAC Setup wird ebenfalls die Clock als Triggersignal ausgewählt und der DAC1 als Ausgang festgelegt.

Grundlegend wird das Timing nun über das Zusammenspiel des Timers mit dem ADC-Handler geregelt. Dieser wird ausgeführt, sobald der ADC durch den Timer getriggert wird, also ebenfalls mit der gewählten Abtastfrequenz. Daher ist es sehr wichtig, dass die Zeit, die zum Durchlaufen des ADC-Handlers benötigt wird, deutlich kleiner ist, als die Dauer, um einen Abtastwert zu wandeln.

Wie bereits in Kapitel 2.3.2 festgelegt wurde, müssen sich die Abtastraten für den ADC und den DAC jeweils unterscheiden. Der Arduino bietet insgesamt drei Timer-Clocks mit jeweils drei Kanälen (Atmel 2015: 857), daher könnte einfach ein zweiter Timer mit der entsprechend anderen Abtastfrequenz konfiguriert werden. Allerdings werden sowohl die A/D- als auch die D/A-Wandlung im ADC-Handler ausgeführt. Dieser wird jedoch durch den ADC getriggert, somit wäre auch durch einen zweiten Timer keine Unabhängigkeit gegeben. Das Problem lässt sich jedoch lösen, indem auf Senderseite nicht jeder Wert in einen Buffer gespeichert wird. Ähnlich verhält es sich auf Empfängerseite. Hier wird zwar jeder Eingangswert gespeichert, aber nicht jedes Mal ein Wert ausgegeben.

Die Zeiten sind abhängig vom gewählten Schlüssel, wie es bereits in Kapitel 2.3.2 erklärt wurde.

### 3.3.2 Synchronisierung

#### Chiffrierer

```

void loop()
{
  if(frameCount==key){ //ist zu Beginn immer der Fall, später zählt der frameCount bis zur Blocklänge hoch
    while (bark<8){
      val = b[bark]; // übergibt ersten Wert der Barker-Codesequenz an outputBuffer
      outputBuffer[obFront]=val;
      bark++; // bewegt Pointer für Barker-Codesequenz
      obFront=(obFront+1)*OUTPUTSIZE; // bewegt Pointer für outputBuffer
      obCount++; // zählt Werte, die in den outputBuff übergeben wurden
    }
    frameCount=0; // Startwert für frameCount
  }
  bark=0; // setzt Pointer für Barker-Codesequenz wieder auf erste Array-Position
  if(ibCount!=0){ // prüft, ob Werte im inputBuffer sind
    val=inputBuffer[ibBack];
    cryptBuffer[w*(key-1)]=val; // übergibt den Wert in den cryptBuffer, der Pointer w wird durch die XOR-Operation rückwärts bewegt
    if (w!=CRYPTMASK){ //immer wahr, außer, wenn w an letzter Position des Buffers ist
      w=(w+jump)&CRYPTMASK; //bewegt Pointer je nach Schlüssel an unterschiedlichen Positionen
      check=w*CRYPT;
      if (check<jump){ //sorgt dafür, dass sich der Pointer nicht im Kreis dreht
        w=(w+1)&CRYPTMASK; //bewegt den Pointer eine Position weiter
      }
    }
  }
}

```

**Abbildung 3.4:** Programmierung der Synchronisierung des Chiffrierers

Auf Senderseite soll vor jedem Block die Barker-Codesequenz gesendet werden. Dafür muss der Loop grundsätzlich in zwei kleinere Loops unterteilt werden, wobei im ersten die Barker-Codesequenz in den outputBuffer gespeichert wird und im zweiten Werte verschlüsselt und übergeben werden. Die Loops müssen jeweils so lange durchlaufen werden, wie die jeweiligen Blöcke lang sind.

Realisiert werden diese durch zwei Verzweigungen, deren Bedingungen so gewählt sind, dass sie nicht gleichzeitig erfüllt werden können. Hierfür gibt es die Variable „frameCount“, die zu Beginn den Wert des Schlüssels hat. So wird die erste if-Bedingung erfüllt, damit anschließend die Barker-Codesequenz mit Hilfe einer While-Schleife an den outputBuffer übergeben werden kann. Die Bedingung der While-Schleife ist für acht Schleifendurchgänge erfüllt, was der Länge der gesendeten Barker-Codesequenz entspricht. Die Werte der Codesequenz sind zu Beginn in einem Array gespeichert, auf das nun mit Hilfe eines Pointers („bark“) zugegriffen wird. Nach Durchlaufen der Schleife wird die Variable „frameCount“ wieder auf Null gesetzt, sodass die Bedingung der nächsten Verzweigung erfüllt wird. In dieser findet nun die Verschlüsselung statt.

Innerhalb dieser Verzweigung zählt die Variable „frameCount“ mit jedem Durchlaufen um Eins hoch, sodass bei Erreichen des Wertes des Schlüssels wieder ausschließlich die erste Verzweigung erfüllt wird und nun vor dem nächsten Block wieder die Barker-Codesequenz ausgegeben wird.

Die Werte der Barker-Codesequenz sind die Minimal- und Maximalwerte, die der Arduino erhalten kann, bei einer Bittiefe von 12 Bit entspricht das  $2^{12} = 4096$  Wer-

ten. Der Wertebereich des Arduino geht also von 0 bis 4095, daher ist die Barker-Codesequenz angepasst und enthält nun statt -1 den Wert 0 und statt 1 den Wert 4095.

## Dechiffrierer

```

void loop()
{
  if(frameCount==key){ //ist zu Beginn immer der Fall, später zählt der frameCount bis zur Blocklänge hoch
    while (bark<8){
      val = b[bark]; // übergibt ersten Wert der Barker-Codesequenz an outputBuffer
      outputBuffer[obFront]=val;
      bark++; // bewegt Pointer für Barker-Codesequenz
      obFront=(obFront+1)%OUTPUTSIZE; // bewegt Pointer für outputBuffer
      obCount++; // zählt Werte, die in den outputBuff übergeben wurden
    }
    frameCount=0; // Startwert für frameCount
  }
  bark=0; // setzt Pointer für Barker-Codesequenz wieder auf erste Array-Position
  if(ibCount!=0){ // prüft, ob Werte im inputBuffer sind
    val=inputBuffer[ibBack];
    cryptBuffer[w*(key-1)]=val; // übergibt den Wert in den cryptBuffer, der Pointer w wird durch die XOR-Operation rückwärts bewegt
    if (w!=CRYPTMASK) { //immer wahr, außer, wenn w an letzter Position des Buffers ist
      w=(w+jump)&CRYPTMASK; //bewegt Pointer je nach Schlüssel an unterschiedlichen Positionen
      check=w%CRYPT;
      if (check<jump) { //sortiert dafür, dass sich der Pointer nicht im Kreis dreht
        w=(w+1)&CRYPTMASK; //bewegt den Pointer eine Position weiter
      }
    }
  }
}

```

**Abbildung 3.5:** Programmierung der Synchronisierung des Dechiffrierers

Auf der Seite des Empfängers muss es nun ähnlich wie auf Senderseite zwei Loops geben, die abwechselnd durchlaufen werden. Im ersten Loop muss die Barker-Codesequenz durch Korrelation erkannt werden, sodass anschließend die Werte eines Blocks entschlüsselt und ausgegeben werden können. Wie oft der Loop durchlaufen werden muss, ist also unklar. Daher wurde die Bedingung so gewählt, dass sie so lange wahr ist, bis der Blockbeginn erkannt wurde. Dies wird wieder über eine Variable realisiert, die bei jedem Durchlauf hochzählt. Wenn das Faltungsergebnis einen eingestellten Schwellenwert überschreitet, soll der zweite Loop wieder so lange durchlaufen werden, wie der Block lang ist. Hier findet dann die Dechiffrierung statt und Übergabe der Werte an den Ausgangsspeicher statt.

Bei der Autokorrelation handelt es sich um eine Faltung des Signals mit sich selbst, bei der keines der Signale im Gegensatz zur Faltung invertiert wird (Görne 2011: 139). Die diskrete Faltung berechnet sich nach folgender Formel:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n - k] \quad (3.2)$$

Damit die Faltung Ergebnisse wie in der Vorberechnung (2.3.1) liefert, müssten die Werte entweder 1 oder -1 betragen. Daher werden die Eingangswerte mit 2047 subtrahiert, sodass der Arbeitswertebereich nun von -2048 bis +2047 reicht. Dementspre-

chend sind auch die auf Empfängerseite gespeicherten Werte der Barker-Codesequenz gewählt.

Nach der Formel 3.2 wird nun das Eingangssignal mit der Barker-Codesequenz gefaltet, die auch hier in einem Array gespeichert ist. Genau genommen findet deshalb keine Autokorellation statt, sondern kann als Filter betrachtet werden, wobei die Werte der Barker-Codesequenz den Koeffizienten entsprechen. Nach jeder Berechnung verweist der Pointer „frame“ auf den nächsten Wert, der im inputBuffer gespeichert ist. So schiebt sich das Eingangssignal nun durch das Filter. Wenn das Eingangssignal so verschoben wurde, dass es genau der Barker-Codesequenz entspricht, liegt das Faltungsergebnis bei seinem Maximum. Tritt dieser Fall ein, werden die Variablen so gesetzt, dass der zweite Loop von nun an durchlaufen wird. Außerdem wird der Pointer „frame“ um die Blocklänge weiterbewegt, sodass er jetzt dort steht, wo die nächste Barker-Codesequenz eintreffen sollte.

#### **Einstellung des Schwellenwertes für die Barker-Codesequenz**

Da auch die Barker-Codesequenz über einen Kanal übertragen wird und somit deren Einflüssen ausgesetzt ist, muss der Schwellenwert für die Erkennung entsprechend niedriger gesetzt werden als der in der Theorie höchstmögliche Wert. Durch die Skalierung der Werte liegt der theoretische Höchstwert bei 29343748. Verbindet man den Ausgang des Chiffrierers direkt mit dem Eingang des Dechiffrierers, liegen die Minimal- und Maximalwerte nicht mehr bei 0 und 4095, sondern bei  $0,55 V \frac{4096}{3,3V} = 683$  und  $2,75 V \frac{4096}{3,3V} = 3413$ . Dies liegt an der geringeren Spannungsabgabe des DAC. Da allerdings das Signal über die direkteste Verbindung den wenigsten Kanaleinflüssen ausgesetzt ist, wird der Schwellenwert so gewählt, dass die Synchronisierung auch so funktioniert (siehe Abbildung 3.6).

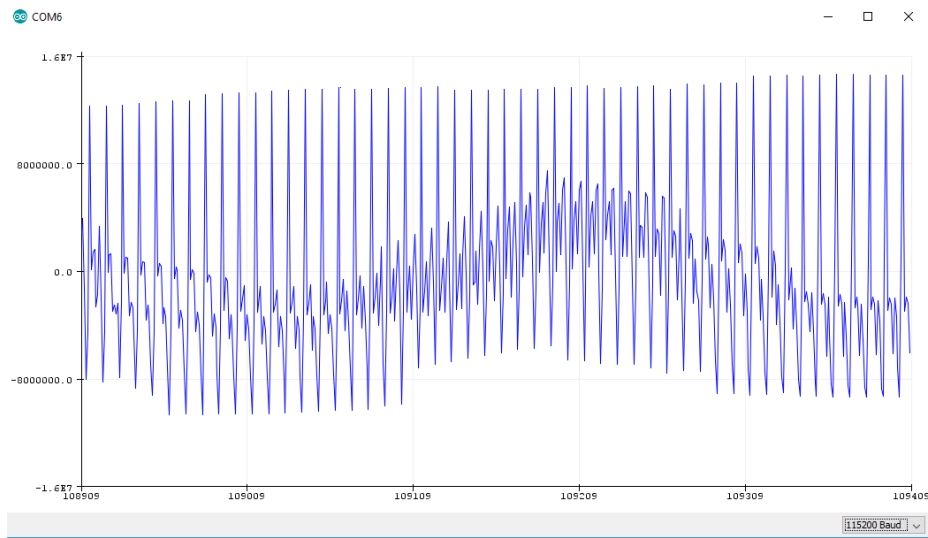
Die Tatsache, dass ein Schwellenwert eingestellt werden muss und keine echte Autokorrelation stattfindet, ist nicht ideal. So kann sich das System nicht selbst für unterschiedliche Kanäle konfigurieren. Besser wäre die Umsetzung über eine tatsächliche Autokorrelation, das heißt, eine Faltung mit den Werten, die auch zuvor im Eingangsspeicher gespeichert wurden. Anstelle eines eingestellten Schwellenwertes könnte das System selbst das Faltungsergebnis mit dem vorherigen Ergebnis vergleichen und so den Framebeginn erkennen.

#### **3.3.3 Chiffrierung und Dechiffrierung**

Die Verschlüsselung wird wie in Kapitel 2.2.3 geplant, umgesetzt. Hierfür wird zunächst die Speicherstruktur erklärt (Abbildung 3.7).

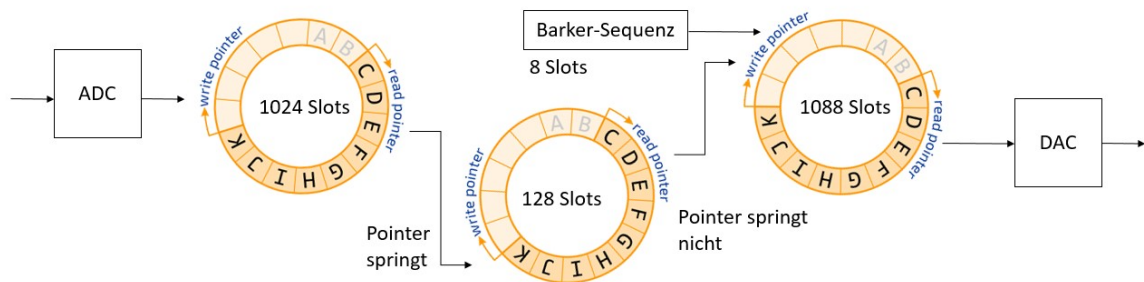
Das Eingangssignal wird zunächst im ADC-Handler gewandelt und anschließend in den inputBuffer gespeichert. Der Pointer „ibFront“ bewegt sich dabei, sodass die Werte an unterschiedlichen Stellen gespeichert werden. Der Speicher ist als Ringbuffer umgesetzt, sodass die Werte, die zuerst hineingeschrieben wurden, wieder überschrieben werden, wenn der Pointer diese Position wieder erreicht. Umgesetzt wird

### 3 Umsetzung



**Abbildung 3.6:** Screenshot des seriellen Plotters, Ausgabe der Variable y zur Einstellung des Schwellenwertes

#### Buffer-Struktur auf Senderseite am Beispiel Key = 128



**Abbildung 3.7:** Buffer-Struktur auf Senderseite am Beispiel des Schlüssels 128

dies durch eine AND-Operation mit der BUFMASK, wenn die Speichergröße dies erlaubt. Diese Rechnung liefert allerdings nur die gewünschten Ergebnisse, wenn die BUFMASK so gewählt ist, dass ihr binärer Wert nur aus Einsen besteht. Ansonsten erfolgt der Übergang des Pointers von der letzten auf die erste Speicherposition mittels einer Modulo-Rechnung mit dem Wert der Buffergröße. Zusätzlich dazu gibt es einen outputBuffer, in den die Werte übergeben werden. Dafür wird für den inputBuffer ein zweiter Pointer benutzt, „ibBack“, der sich nach dem gleichen Schema weiterbewegt. So können die Werte unabhängig voneinander geschrieben und gelesen werden. Das gleiche Prinzip wird auch für den outputBuffer verwendet. Die Verschlüsselung selbst geschieht im cryptBuffer.

Der Pointer dieses Buffers bewegt sich anders als alle anderen nicht um einen Wert, sondern um eine bestimmte Anzahl an Werten. Auf Senderseite liegt dieser Wert immer bei 16, auf Empfängerseite variiert er je nach Schlüssel, sodass die gedachte  $m \times n$ -Matrix insgesamt entsprechend viele Elemente besitzt. Um den Übergang des Pointers an Zeilenenden zu ermöglichen, wie in Abbildung 2.4 dargestellt, wird durch eine Modulo-Rechnung geprüft, ob das Ergebnis 0 ist. Für diesen Fall wird der Pointer nur um eine Stelle weiterbewegt. Diese Berechnung funktioniert jedoch nicht für den Übergang von der letzten auf die erste Position, daher wird für diesen Fall eine zusätzliche Verzweigung eingefügt, deren Bedingung prüft, ob die letzte Position erreicht wurde.

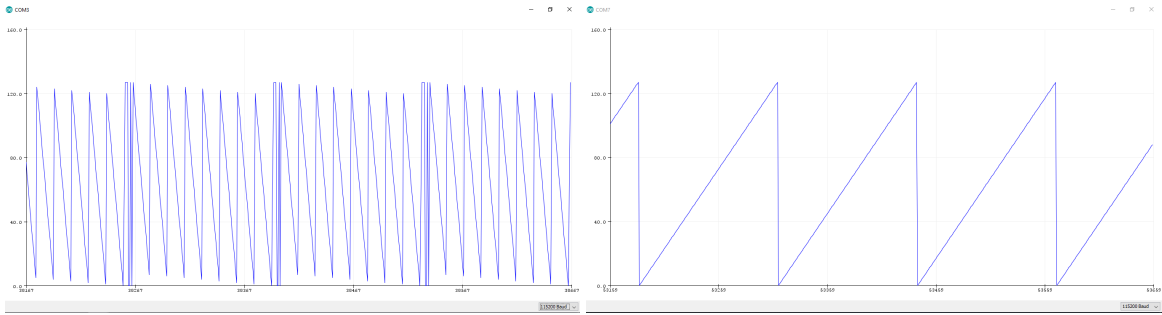
Die Verschlüsselung erfolgt also nur über die Bewegung des Pointers und damit das Schreiben der Werte nach der Verschlüsselungsstruktur in den Speicher. Dieser wird nun allerdings wie auch der Eingangs- und Ausgangsspeicher ausgelesen. Das führt dazu, dass beim allerersten Durchgangs an die ersten Positionen noch kein Wert geschrieben wurde, diese aber schon ausgelesen wird. Beim zweiten Block werden dafür die zuvor dorthin geschriebenen Werte ausgelesen, sie gehen also nicht verloren. Der Vorteil dabei ist, dass durchgehend Werte an den Ausgang gesendet werden können und nicht vor jedem Block gewartet werden muss, bis die Werte des gesamten Blocks umstrukturiert wurden.

Die Abbildung 3.8 zeigt die Verschlüsselung voreingestellter Werte graphisch. Die Barker-Codesequenz wurde für eine bessere Sichtbarkeit anders skaliert.

Es ist gut zu erkennen, dass die Chiffrierung nach dem geplanten Muster funktioniert. Der Empfänger entschlüsselt das Signal auf die identische Weise. Der einzige Unterschied liegt in den Sprüngen des Pointers, denn wenn die verwendete Matrix nicht quadratisch ist, dürfen diese nicht gleich sein. Daher berechnet sich der Sprungwert in Abhängigkeit des Schlüssels.

Auch hier kann exemplarisch gezeigt werden, dass dies funktioniert, indem die Eingangswerte, inklusive der Barker-Codesequenz, schon direkt im Sketch gespeichert werden. In Abbildung 3.8 ist auf der rechten Seite das entschlüsselte Signal zu erkennen. Auch kann hierdurch gezeigt werden, dass die Synchronisierung mit der Barker-Codesequenz funktioniert, da diese nicht an den Ausgang gesendet wurde.

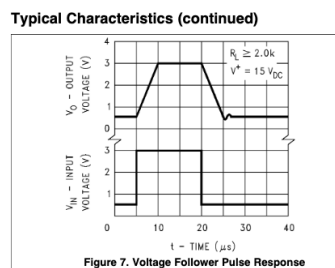
### 3 Umsetzung



**Abbildung 3.8:** links: exemplarische Chiffrierung mit skaliertem Barker-Codesequenz, rechts: exemplarische Dechiffrierung

## 3.4 Probleme und Verbesserungsmöglichkeiten

Grundsätzlich treten bei der verwendeten Schaltung allerdings noch Probleme auf. Es zeigt sich, dass die Operationsverstärker LM358N nicht die ideale Wahl für die Schaltung sind.

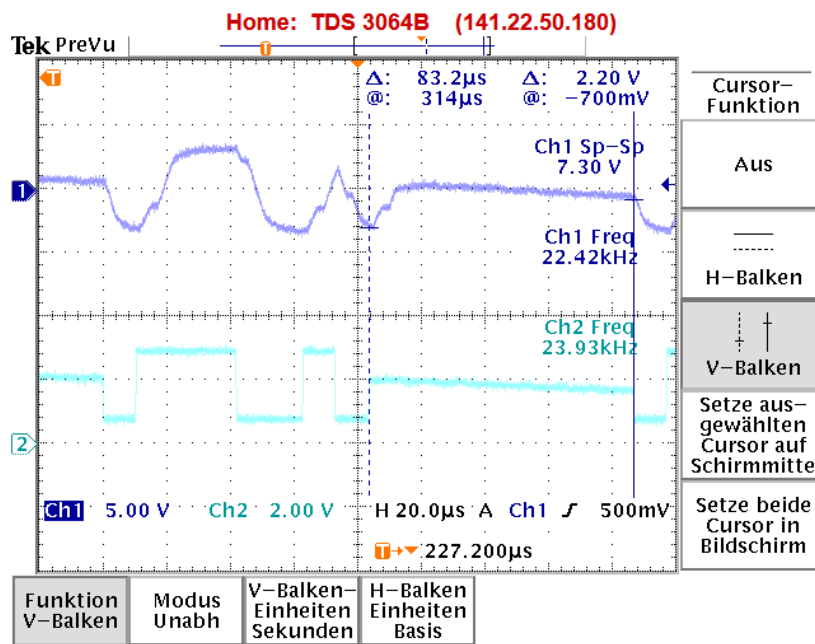


**Abbildung 3.9:** Voltage Follower Pulse Response des LM358-N ([Texas Instruments 2014](#))

Das Problem liegt an der zu geringen Spannungsanstiegsrate. Abbildung 3.9 wird dieses Problem deutlich, da nur für den Spannungsanstieg von 3 V 10  $\mu$ s benötigt werden. Daher schaffen es die verbauten Operationsverstärker nicht, die Barker-Codesequenz ohne Verzögerung zu übertragen. Dadurch kann diese zwar ohne Probleme am Empfänger erkannt werden, allerdings werden die durch den LM358N verzögerten Werte als Teil des verschlüsselten Blockes interpretiert und führen so zu Signalstörungen. Die Abtastrate beträgt auf Bild 3.10 allerdings 96 kHz, sodass das Problem noch deutlicher wird als bei der verwendeten Abtastrate von 30 kHz. Es ist allerdings gut zu erkennen, dass die Barker-Codesequenz nach der Schaltung am Output in ihrer Länge verzögert ist, während die Zeit für Kanal 2 dem berechneten Wert von  $\frac{1}{96000\text{Hz}} \cdot 8 = 83,3\mu\text{s}$  entspricht.

Dem kann Abhilfe geleistet werden, indem der Chiffrierer direkt über den DAC-Output an den ADC-Input des Dechiffrierers angeschlossen wird. Hierfür gibt es am

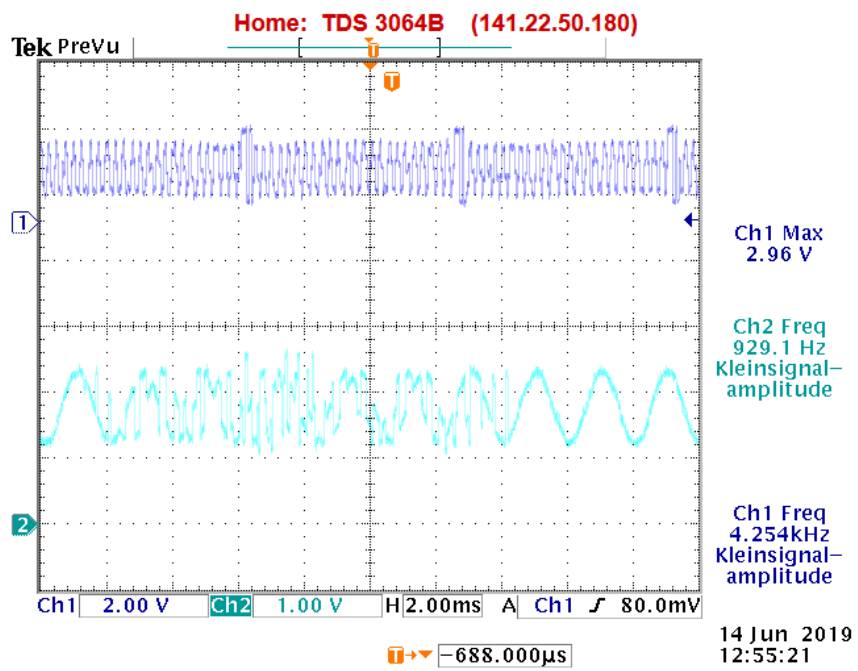
### 3 Umsetzung



**Abbildung 3.10:** Barker-Codesequenz am Output der Schaltung (Kanal 1) und direkt am DAC (Kanal 2)



### 3 Umsetzung



**Abbildung 3.11:** Signalstörung nach Auseinanderlaufen der Timer, Kanal 1: DAC des Chiffrierers, Kanal 2: DAC des Dechiffrierers

Dechiffrierer einen Schalter, der die Schaltung zwischen R5 und dem ADC-Input unterbricht (siehe Schaltung im Anhang A.3). Dies hat allerdings den Nachteil, dass das Ausgangssignal des DAC nicht mehr durch die Differenzverstärkerschaltung läuft und daher am Dechiffrierer keine Vollaussteuerung erreicht werden kann.

Eine bessere Variante wäre daher die Verwendung anderer Operationsverstärker. Hierfür würde sich beispielsweise der TSH112 eignen, dessen Hauptanwendungszweck die Videosignalübertragung ist. Da im Videobereich deutlich höhere Frequenzen übertragen werden müssen, liegt die Grenzfrequenz des TSH112 nicht wie die des LM358N bei 1MHz (Texas Instruments 2014), sondern beträgt 100MHz (ST Microelectronics 2002). Die Slew Rate<sup>8</sup> des TSH112 liegt bei 1  $\mu$ s pro 45 V, somit wäre der Operationsverstärker in der Lage, die Barker-Codesequenz korrekt zu übertragen.

Auch die Programmierung lässt sich weiter verbessern. Ein großer Nachteil ist noch der komplizierte Schlüsselaustausch. Momentan müssen dafür sowohl der Sende- als auch der Empfänger mit einem PC oder Laptop, auf dem eine Programmierumgebung für den Arduino Due installiert ist, verbunden werden. In den jeweiligen Sketchen muss nun der Schlüssel umgestellt werden. Dies ist für einen potentiellen Nutzer nicht zweckmäßig und zudem nicht sicher. Im Idealfall kennt nicht einmal der Nutzer selbst den verwendeten Schlüssel, indem dieser zufällig ermittelt wird.

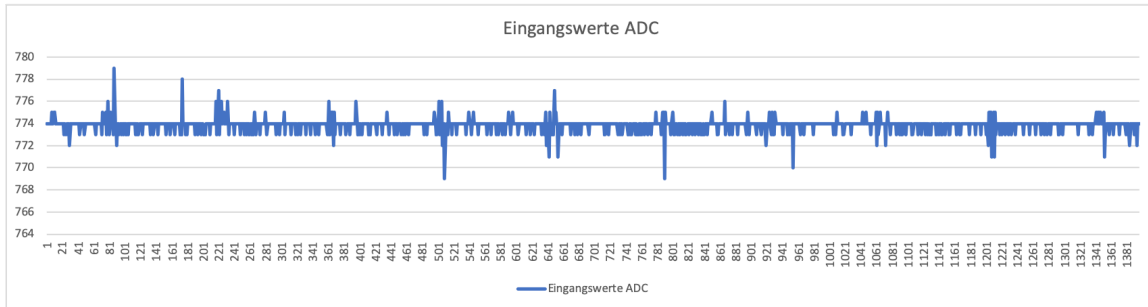
Das Problem der umständlichen Bedienbarkeit ließe sich für die Weiterentwicklung des Prototypen beispielsweise über die Eingabe in eine Bedienoberfläche oder ein Nummernfeld mit anschließend grün aufleuchtender LED als Bestätigung für eine gültige Eingabe lösen. Die Schlüssel könnten ebenfalls sehr einfach umbenannt werden, in 1, 2, 3, 4, sodass der Nutzer nicht wissen muss, welche Schlüssel tatsächlich verwendet werden.

Die deutlich bessere Lösung wäre allerdings die Realisierung einer Infrarot-Schnittstelle, wie sie auch bei vorhandenen Produkten zum Schlüsselaustausch genutzt wird. Außerdem müsste der Schlüssel selbst möglichst zufällig gewählt werden. Durch die unterschiedlichen Latenzen ist es für einen potenziellen Mithörer jedoch trotzdem relativ einfach, herauszufinden, welcher Schlüssel verwendet wurde und wie viele verschiedene Schlüssel es gibt. Es wäre also eine weitere Idee zur Weiterentwicklung, den Einfluss des Schlüssels nicht mehr nur auf die gewählte Blocklänge zu beschränken. Eine Idee könnte beispielsweise sein, die Reihenfolge der ausgegebenen Spalten jeweils neu festzulegen. Hierfür müsste jedoch sichergestellt werden, dass immer eine ausreichende Sprachverständlichkeit gegeben ist.

---

<sup>8</sup>deutsch: Spannungsanstiegsrate

### 3 Umsetzung



**Abbildung 3.12:** Messung der Eingangswerte des ADC bei Anlegen einer konstanten Spannung in Höhe von 0,6 V

Ein weiteres Problem ist die Ungenauigkeit der Wandler der Arduino Due. Bei einer Bittiefe von 12 und damit 4096 Quantisierungsstufen, die unter einem maximal möglichen Spannungsbereich von 0 V bis 3,3 V aufgeteilt werden, beträgt der Spannungsunterschied zwischen zwei Quantisierungsstufen nur  $805 \mu\text{V}$ . Wird eine konstante Spannung an den Eingang angelegt, sollte diese im Idealfall auch konstant auf den selben Wert gewandelt werden. Dies ist jedoch nicht der Fall, wie Abbildung 3.12 zeigt. Daher wäre eine Überlegung, als Verbesserung andere Wandler zu verwenden, die entweder das Signal genauer wandeln können, oder in einem höheren Spannungsbereich angesteuert werden können, sodass der Abstand zwischen den einzelnen Quantisierungsstufen größer wird.

# 4 Validierung

## 4.1 Durchzuführende Messungen

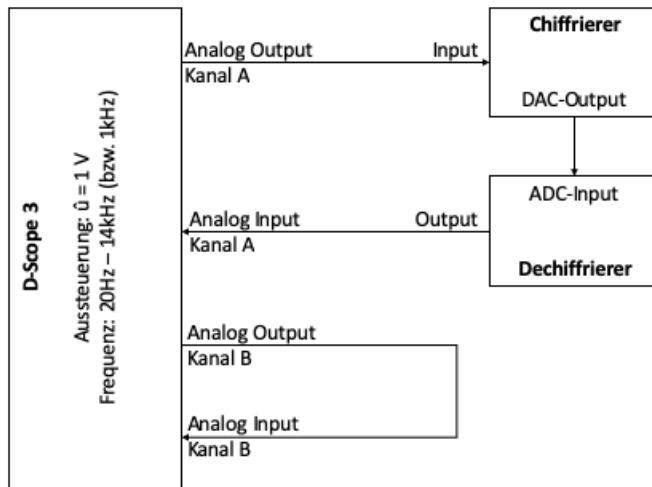
Um zu prüfen, ob die gesetzten Ziele 1.1 erreicht wurden, ist eine messtechnische Überprüfung nötig. Die dabei interessantesten Messungen sind die Phasenfrequenzgänge der einzelnen Systemkomponenten bei der Verwendung verschiedener Schlüsse. Aus den Phasenfrequenzgängen lassen sich die Gruppenlaufzeiten, also die Latenzen, berechnen und mit dem eingangs gesetzten Ziel und den Vorberechnungen vergleichen. Darüber hinaus sollen die Amplitudenfrequenzgänge und der Klirrfaktor gemessen werden, um Informationen über die Qualität des verschlüsselten Signals zu erhalten.

Die Messungen werden mit dem Messsystem D-Scope 3 der Firma Prismsound durchgeführt. Für die Messungen der Einzelkomponenten wird jeweils der analoge Output des D-Scope 3 an den Input und der Output wieder an den analogen Input des D-Scope 3 angeschlossen. Der Mic-Input des Chiffrierers wird nur für die Messung des Amplitudenfrequenzgangs genutzt (Abbildung 4.1). Für die Messungen mit unterschiedlichen Schlüsseln werden Chiffrierer und Dechiffrierer direkt über den DAC-Output und den ADC-Input verbunden. In der Software wird die Ausgangssignalspannung mit einem Spitzenwert von 1 V eingestellt<sup>1</sup>. Der Frequenzbereich der Messungen reicht von 20 Hz bis 14000 Hz. Diese Einstellungen entsprechen nicht der DIN-Norm (DIN EN 61606-3 2009), da eigentlich mit einem Pegel von -20dBFS und einem Frequenzbereich von 20 Hz bis zur halben Abtastfrequenz gemessen werden müsste. Allerdings variiert die Eingangsabtastfrequenz je nach Schlüssellänge (2.3.2), daher wurde eine obere Messgrenzfrequenz von 14000 Hz gewählt. Für die Messungen der Phasenfrequenzgänge ist dieser Aufbau identisch, allerdings werden zusätzlich der analoge Ein- und Ausgang von Kanal B des D-Scope-Messsystems direkt über ein Kabel verbunden, sodass die Phase zwischen beiden Kanälen gemessen werden kann. Auch für die Messungen der Klirrfaktoren (THD+N)<sup>2</sup> bleibt der Messaufbau identisch. Allerdings variiert nun die Aussteuerung, während die Frequenz für alle Messungen 1 kHz beträgt.

---

<sup>1</sup>Außer anders angegeben

<sup>2</sup>Total Harmonic Distortion + Noise



**Abbildung 4.1:** Messaufbau für Messungen mit dem D-Scope 3 der Firma Prismsound

## 4.2 Amplitudenfrequenzgang

Bei der Messung des Amplitudenfrequenzgangs wird zum Einen erwartet, dass die Herstellerangaben eines linearen Frequenzgangs von 45 Hz bis 15000 Hz  $\pm 2$  dB bestätigt werden (Shure 2010) und zum Anderen, dass bei den Messungen für Chiffrierer und Dechiffrierer der Einfluss der Tiefpassfilter sichtbar wird. Die Verwendung unterschiedlicher Schlüssel sollte idealerweise den Verlauf des Amplitudenfrequenzgangs nicht beeinflussen.

Diese Thesen werden durch die Messungen weitestgehend bestätigt. Es ist allerdings festzustellen, dass auch tiefe Frequenzen (Messungen des Verschlüsselungssystems) stark bedämpft werden. Der lineare Bereich reicht je nach Messung von 200 Hz bis 5000 Hz  $\pm 2$  dB (Chiffrierer ohne Verschlüsselung) bis zu 300 bis 7000 Hz  $\pm 2$  dB (Dechiffrierer ohne Verschlüsselung).

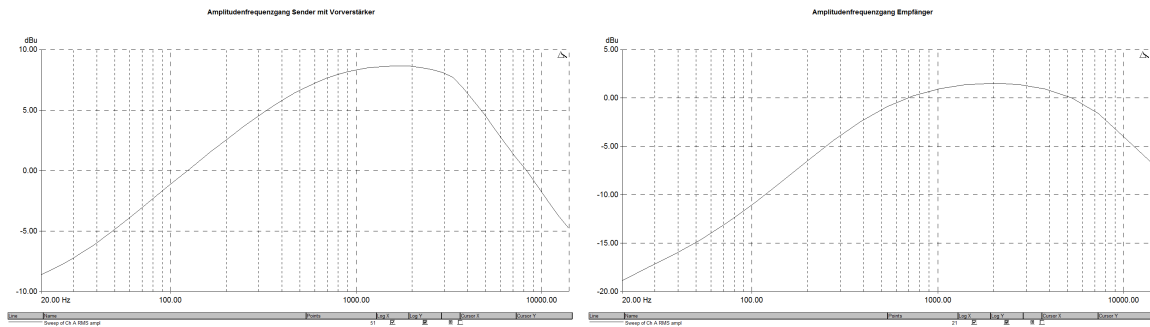
Dies ist insofern verwunderlich, als dass die Grenzfrequenzen der verbauten Tiefpassfilter<sup>3</sup> einen genau umgekehrten Verlauf erwarten lassen. Dies könnte allerdings durch die veränderten Kondensatorwerte zu erklären sein, die neben Störeinflüssen auch das Nutzsignal bedämpfen.

Die Wahl unterschiedlicher Schlüssel wirkt sich dagegen nicht auf die Amplitudenfrequenzgänge aus.

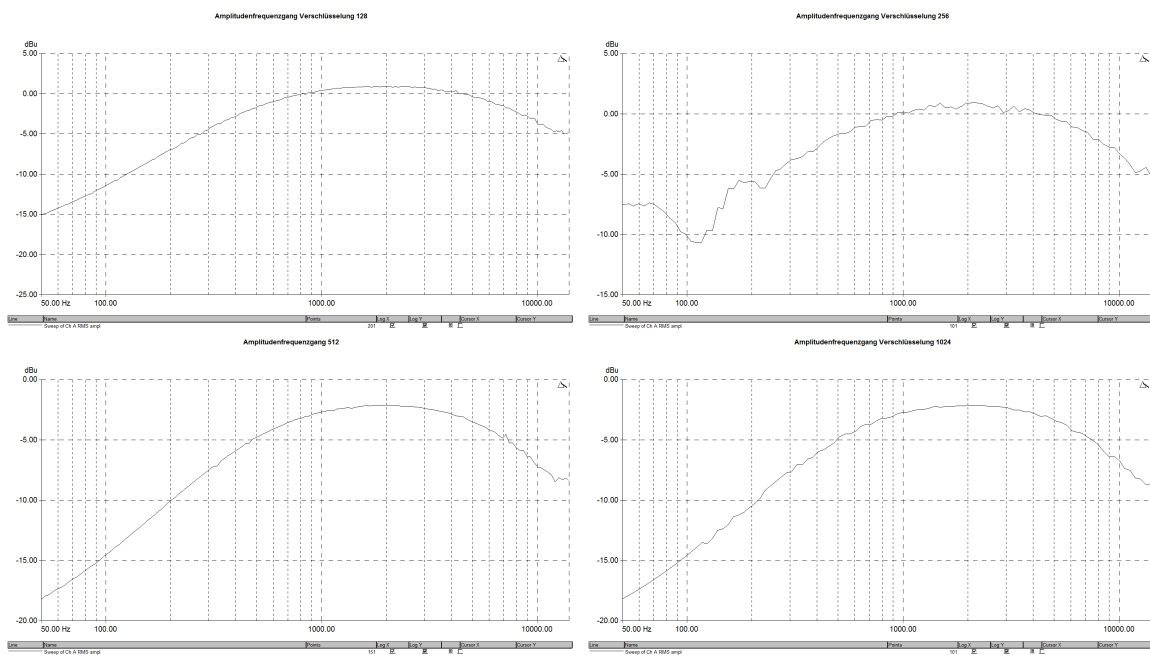
Bei der Messung mit Schlüssel 512 wurde das Signal 2,5dB geringer angesteuert.

<sup>3</sup>Chiffrierer: 10000 Hz, Dechiffrierer: 6000 Hz

## 4 Validierung



**Abbildung 4.2:** links: Amplitudenfrequenzgang des Chiffriers inklusive Mikrofonvorverstärker, rechts: Amplitudenfrequenzgang des Dechiffriers



**Abbildung 4.3:** Amplitudenfrequenzgänge mit unterschiedlichen Schlüsseln

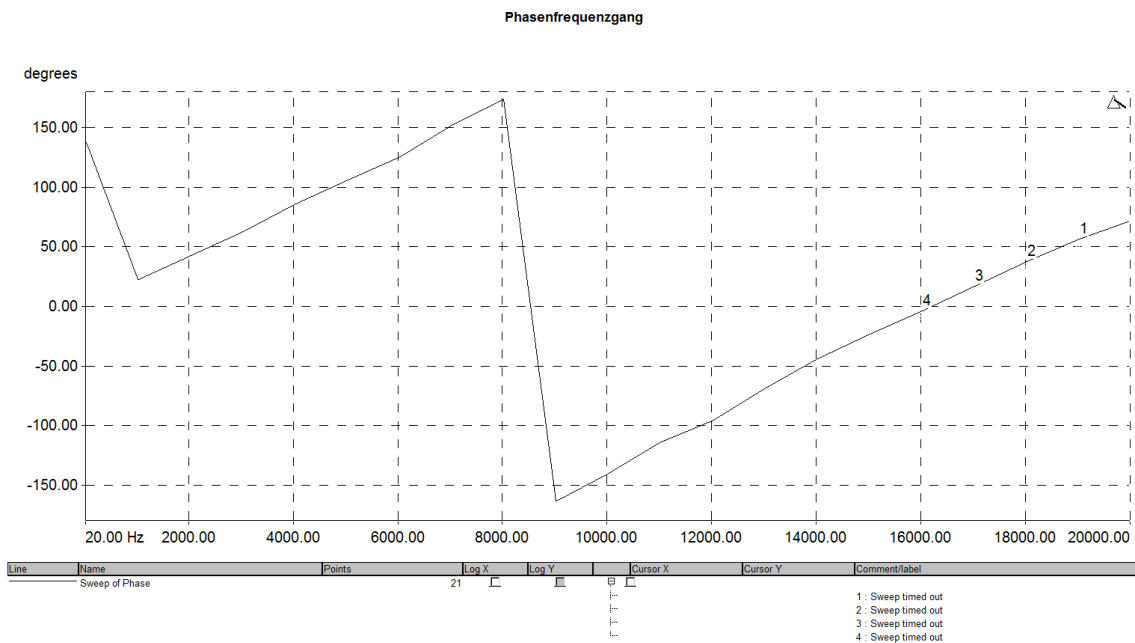
## 4.3 Phasenfrequenzgang

Die Latenz, also die Gruppenlaufzeit, ist die Ableitung der Phasenlaufzeit, sodass sie aus der Steigung des Phasenfrequenzgangs bestimmt werden kann.

Es ist hierbei wichtig zu wissen, dass die Ergebnisse von der Software anders dargestellt werden, als sie eigentlich sind. Die Achsenbeschriftung der Y-Achse reicht dabei von 180 bis -180 Grad, korrekt wäre jedoch eine umgekehrte Beschriftung. Außerdem reichen die Messergebnisse eigentlich über diese Skala hinaus. Dieses Problem der Darstellung von Ergebnissen jenseits der von der Software angebotenen Skala wird durch senkrechte Phasendarstellungssprünge gelöst. Die dargestellten Phasensprünge entsprechen daher also nicht tatsächlich auftretenden Phasensprüngen, sondern sind nur durch die begrenzten Darstellungsmöglichkeiten des Programms sichtbar. An diesen Darstellungsstellen muss daher jeweils eine Phase von 360 Grad subtrahiert werden.

### 4.3.1 Phasenfrequenzgang PGX-System

Die Messung des Phasenfrequenzgangs wird zunächst für die verwendete Mikrofonfunktstrecke ohne Chiffrierer und Dechiffrierer durchgeführt.



**Abbildung 4.4:** Shure PGX: Phasenfrequenzgang

Der Graph des Phasenfrequenzgangs des PGX-Systems zeigt einen Phasendarstellungssprung, der jedoch nicht wie zuvor erklärt, senkrecht verläuft. Das liegt daran,

dass für die Messung zu wenig Messpunkte eingestellt wurden und der Graph dadurch unterabgetastet wurde. Bei Verwendung von mehr Messpunkten müsste dieser dargestellte Phasensprung wieder senkrecht verlaufen. Trotzdem lassen sich die Messergebnisse verwenden, um daraus die Gruppenlaufzeit zu berechnen. Da diese die Ableitung der Phasenlaufzeit ist, lässt sich diese Berechnung durch zwei Punkte, die ein Steigungsdreieck bilden, durchführen.

$$\tau_{Gruppe} = \frac{(\varphi_2 - \varphi_1)}{360^\circ(f_2 - f_1)} \quad [\text{s}] \quad (4.1)$$

Gewählt werden die Frequenzen 2000 Hz und 6000 Hz. Bei 2000 Hz lässt sich eine Phase von  $40^\circ$  ablesen, bei 6000 Hz beträgt diese  $125^\circ$ . Mit der Formel 4.1 errechnet sich eine Gruppenlaufzeit von 59,03µs. Dieser Wert ist für das menschliche Ohr nicht wahrnehmbar.

### 4.3.2 Phasenfrequenzgänge Sender, Empfänger, beide

Betrachtet man die drei Graphen der Phasenfrequenzgänge des Senders, des Empfängers und schließlich von beiden gemeinsam, ist auffällig, dass sie nur an einigen Abschnitten linear verlaufen. Woran das liegt, kann nicht abschließend geklärt werden. Die Software meldet bei der Messung allerdings regelmäßig sogenannte „sweep time outs“. In der Bedienungsanleitung wird erklärt, dass an diesen Punkten kein Messwert ermittelt werden konnte (Dennis 2012: 115). Woran dies liegen kann, wird allerdings nicht erklärt, jedoch könnte der Wert, ab dem das Programm „sweep time outs“ meldet, für zukünftige Messungen erhöht werden.

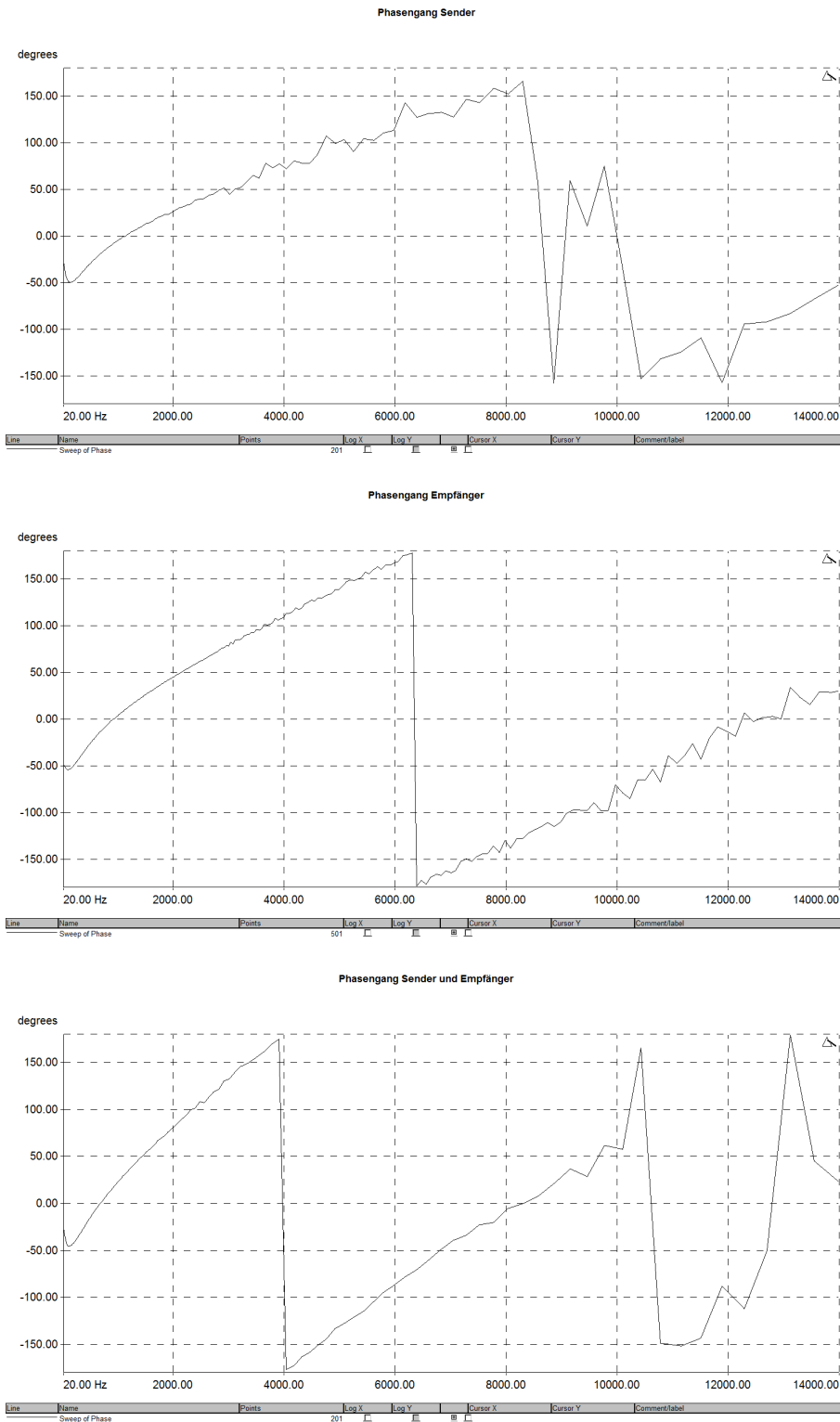
Es ist jedoch tendenziell zu erkennen, wo in den jeweiligen Graphen 4.5 eine gedachte Gerade verlaufen könnte, wenn diese geglättet würden. Daher werden die Punkte für die Berechnung so ausgewählt, dass sie sich nach Möglichkeit auf einem linearen Abschnitt befinden. So entstehen zwar Messungenauigkeiten, allerdings sollten die Ergebnisse für eine erste Einschätzung reichen.

Für die Berechnung der Gruppenlaufzeit des Senders wurden daher die Punkte  $P_1(2000\text{Hz} | -25^\circ)$  und  $P_2(6000\text{Hz} | -110^\circ)$  gewählt. Daraus errechnet sich eine Gruppenlaufzeit von 59,03 µs. Der Graph der Gruppenlaufzeit auf Empfängerseite ist im Bereich von 2000 bis 6000 Hz linear, daher werden die Punkte  $P_1(2000\text{Hz} | -45^\circ)$  und  $P_2(4000\text{Hz} | -110^\circ)$  gewählt. Dies ergibt für den Empfänger eine Gruppenlaufzeit von 90,27 µs. Für die Messung und Berechnung der Gesamtgruppenlaufzeit durch Sender und Empfänger wird nun ein Wert von 149,3 µs erwartet. Die berechnete Gruppenlaufzeit beträgt 138,9 µs, liegt also ca 10 µs neben dem erwarteten Ergebnis. Für die Berechnung wurden folgende Punkte verwendet:  $P_1(2000\text{Hz} | -80^\circ)$  und  $P_2(6000\text{Hz} | -360^\circ + 80^\circ)$ . Die Abweichung lässt sich durch die in den Graphen deutlich erkennbaren nichtlinearen Abschnitte erklären. Bei der Wahl anderer Punkte, lassen sich auch andere Gruppenlaufzeiten berechnen.

Zur Kontrolle wird zusätzlich das Gesamtsystem inklusive Mikrofonsfunkstrecke gemessen, ebenfalls ohne Verschlüsselung. Da die Gruppenlaufzeiten der einzelnen Kom-



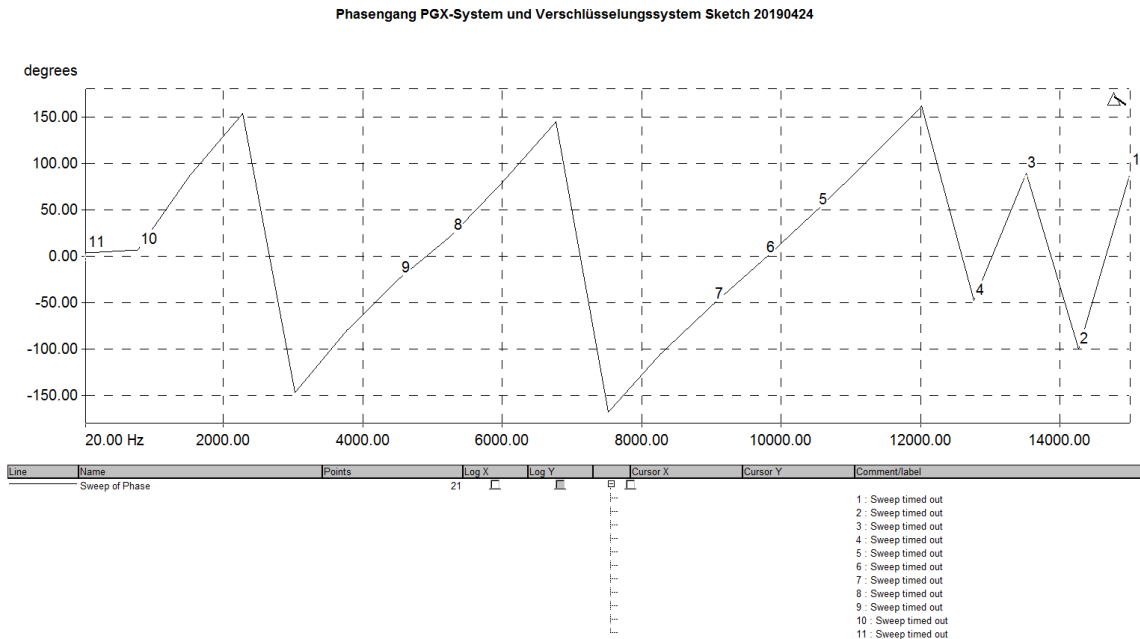
## 4 Validierung



**Abbildung 4.5:** Phasenfrequenzgänge des Chiffrierers (oben), des Dechiffrierers (mitte) und der gesamten Verschlüsselungseinheit (unten)

## 4 Validierung

ponenten bereits ermittelt wurden, lässt sich eine Erwartungswertspanne berechnen. Es werden für diese Berechnungen die Gruppenlaufzeiten für Sender und Empfänger, die jeweils einzeln und gemeinsam gemessen wurden, genutzt und die Gruppenlaufzeit der Mikrofonfunkstrecke hinzuaddiert. Mit  $\tau_{Sender} + \tau_{Empfänger} + \tau_{Funkstrecke} = \tau_{Gesamt}$  beträgt der erwartete Wert  $208,3 \mu s$ . Wird für die selbe Rechnung der gemeinsam ermittelte Wert für  $\tau_{SenderEmpfänger}$  genutzt, wird für die Gesamtgruppenlaufzeit ein Wert von  $197,83 \mu s$  erwartet.



**Abbildung 4.6:** Gesamtsystem: Phasenfrequenzgang

Auch bei der Messung des Gesamtsystems wurden in den Programmeinstellungen zu wenige Messpunkte eingestellt, sodass eine Unterabtastung stattfindet. Zudem sind auch hier einige „sweep time outs“ bei der Messung entstanden.

Wird nun aus diesem Graphen die Gruppenlaufzeit berechnet, erhält man bei Verwendung der Punkte  $P_1(1000Hz | -720^\circ - 20^\circ)$  und  $P_2(1200Hz | -720^\circ - 160^\circ)$  einen Wert in Höhe von  $194,4 \mu s$ . Dieser Wert liegt etwa  $3 \mu s$  unterhalb des unteren Wertes der erwarteten Spanne. Auch hier liegt das aber an der nicht perfekten Linearität des Phasenfrequenzgangs. Dadurch erhält man bei Verwendung anderer Messpunkte auch andere Ergebnisse.

### 4.3.3 Messungen mit Verschlüsselung

Auch hier lässt sich feststellen, dass die Graphen des Phasenfrequenzgänge nur in einigen Abschnitten linear sind (Abbildungen 4.7, 4.8, 4.9 und 4.10). Daher variiert

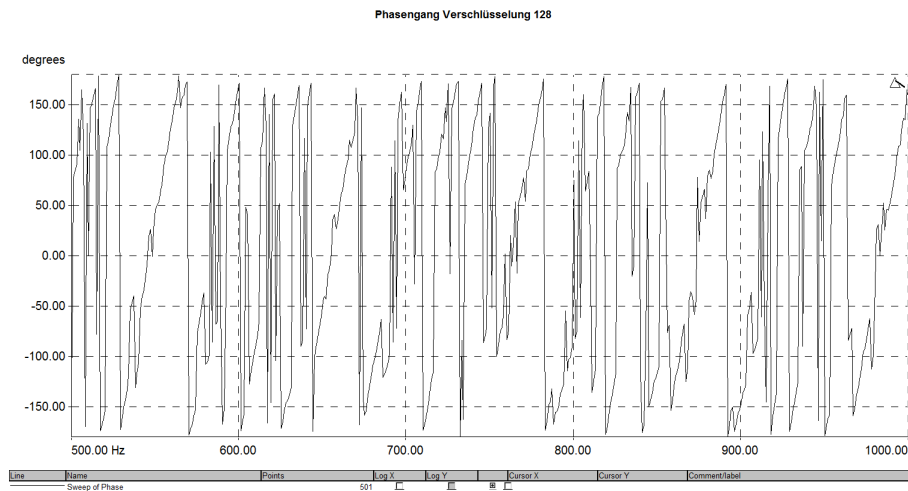
## 4 Validierung

bei diesen Messungen der eingestellte Frequenzbereich. In einigen Bereichen lässt sich gut erahnen, wo vermutlich eine lineare Gerade verlaufen könnte. In andere Bereichen ist dies dagegen kaum möglich.

Eine Erklärung könnte die relativ lange Messdauer sein. Der Verschlüsselungssender und -empfänger haben eine leicht auseinanderlaufende Taktung wodurch regelmäßig Signalstörungen auftreten (siehe Kapitel 3.4). Wenn nun während der Messung bedingt durch eine nicht korrekt erkannte Barker-Codesequenz kein Signal mehr ausgegeben wird, führt dies eventuell zum vorliegenden Verlauf des Graphen. Es kann sein, dass das Messgerät nicht mit diesen Störungen umgehen kann und die Zeit und damit die Phasenverschiebung, bis die gesendete Frequenz wieder ungestört im Messgerät ankommt, daher sehr groß wird. Dies kann allerdings nicht die einzige Erklärung sein, da die Graphen auch schon bei den vorherigen Messungen ähnliche Phänomene aufwiesen, wenn auch nicht in dem Maße.

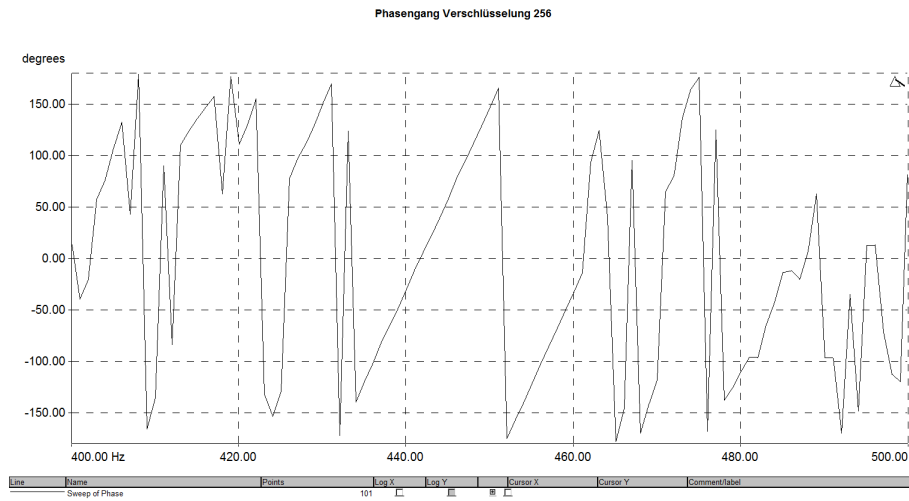
Dennoch soll im Folgenden versucht werden, eine Gruppenlaufzeit aus den vorliegenden Graphen zu berechnen. Es ist klar, dass diese Ergebnisse keine Absolutwerte darstellen und daher nur der groben Orientierung dienen können. Deshalb wird die Gruppenlaufzeit anschließend zusätzlich mit Hilfe eines Oszilloskops gemessen. Auch dies liefert keinen exakten Wert, aber so lässt sich prüfen, ob die Größenordnung mit dem aus dem Phasenfrequenzgang berechneten Ergebnis übereinstimmt.

### Messergebnisse

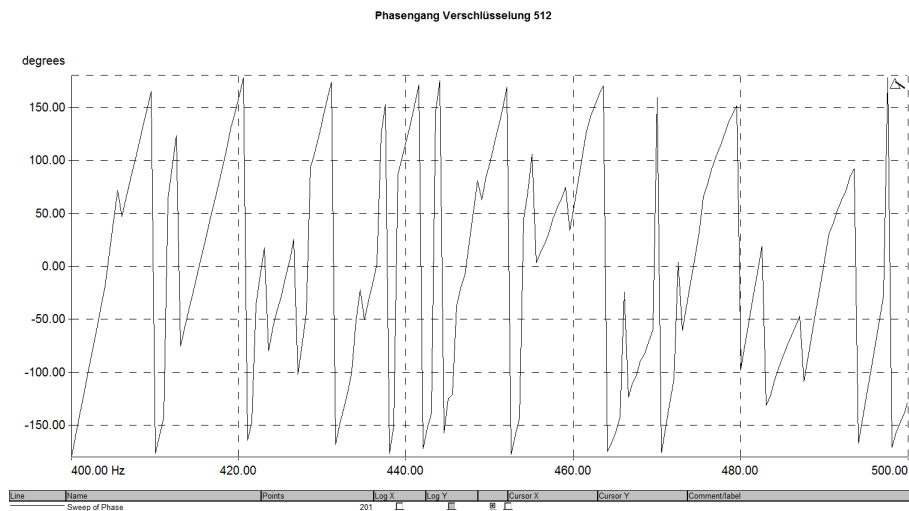


**Abbildung 4.7:** Verschlüsselung 128: Phasenfrequenzgang 500 bis 1000 Hz

## 4 Validierung

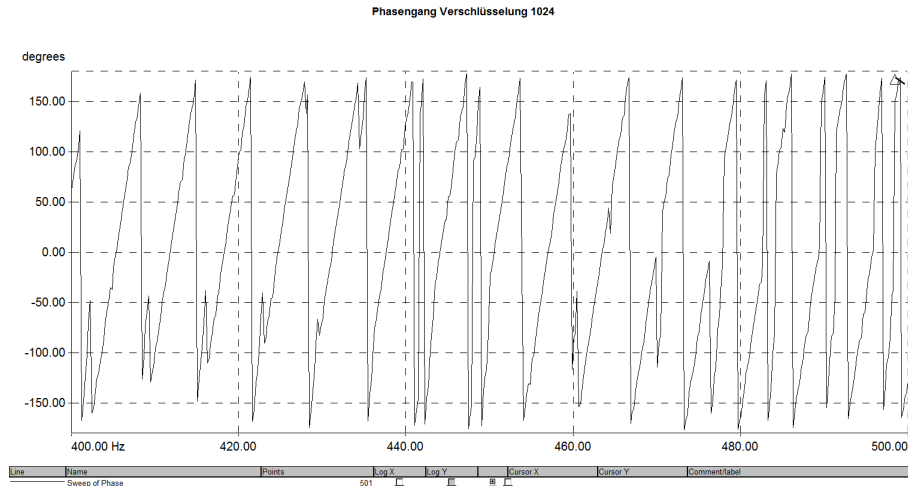


**Abbildung 4.8:** Verschlüsselung 256: Phasenfrequenzgang 400 bis 500 Hz



**Abbildung 4.9:** Verschlüsselung 512: Phasenfrequenzgang 400 bis 500 Hz

## 4 Validierung



**Abbildung 4.10:** Verschlüsselung 1024: Phasenfrequenzgang 400 bis 500 Hz

Bei Verwendung des Schlüssels 128 ergibt sich bei der Berechnung der Gruppenlaufzeit mit Verwendung der Punkte  $P_1 = (550 \text{ Hz} | -360^\circ - 50^\circ)$  und  $P_2 = (580 \text{ Hz} | -720^\circ - 20^\circ)$  ein Wert in Höhe von  $\tau_{Gruppe} = 30,5 \text{ ms}$ . Die Messergebnisse der Messung mit dem Oszilloskop liegen bei 26,4 ms, 30,8 ms und 20,0 ms (4.11).

Wird das Signal mit Schlüssel 256 verschlüsselt, liegt der berechnete Wert bei  $\tau_{Gruppe} = 48,6 \text{ ms}$ . Dieser Berechnung liegen die Punkte  $P_1 = (440 \text{ Hz} | 30^\circ)$  und  $P_2 = (480 \text{ Hz} | -360^\circ + 40^\circ)$  zu Grunde. Wird mit dem Oszilloskop gemessen, liegen die Latenzen bei 23,6 ms, 50,8 ms und 48 ms (4.12).

Bei einem Schlüssel 512 und den verwendeten Punkten  $P_1 = (405 \text{ Hz} | -50^\circ)$  und  $P_2 = (420 \text{ Hz} | -360^\circ - 1600^\circ)$  beträgt die berechnete Gruppenlaufzeit  $\tau_{Gruppe} = 87,03 \text{ ms}$ . Die mit dem Oszilloskop gemessenen Werte (Abbildung 4.13) weichen davon teilweise deutlich ab und liegen bei 46,0 ms, 93,2 ms und 78,8 ms.

Für eine Blocklänge von 1024 berechnet sich aus dem Phasenlaufzeiten  $\varphi_1 = -100^\circ$  der Frequenz 420 Hz  $\varphi_2 = -720^\circ - 120^\circ$ , die bei der Frequenz 440 Hz abgelesen kann eine Gruppenlaufzeit von  $\tau_{Gruppe} = 152,7 \text{ ms}$ . Auch hier weichen die mit dem Oszilloskop gemessenen Werte in Höhe von 80,4 ms, 181,0 ms und 114,0 ms stark vom berechneten Wert ab (4.14).

In der Tabelle 4.1 sind die Ergebnisse noch einmal zusammengefasst:

## 4 Validierung

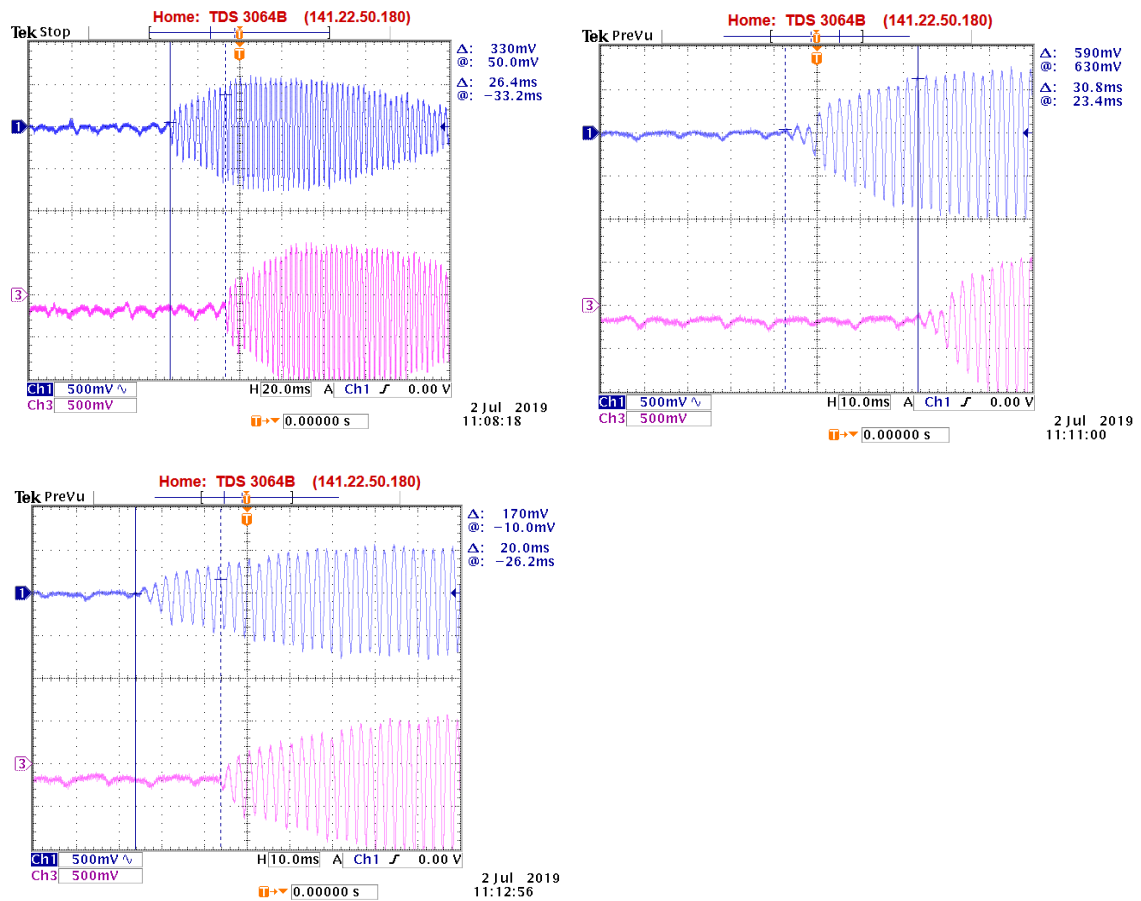


Abbildung 4.11: Messung der Latenz mit Oszilloskop für Schlüssel 128

## 4 Validierung

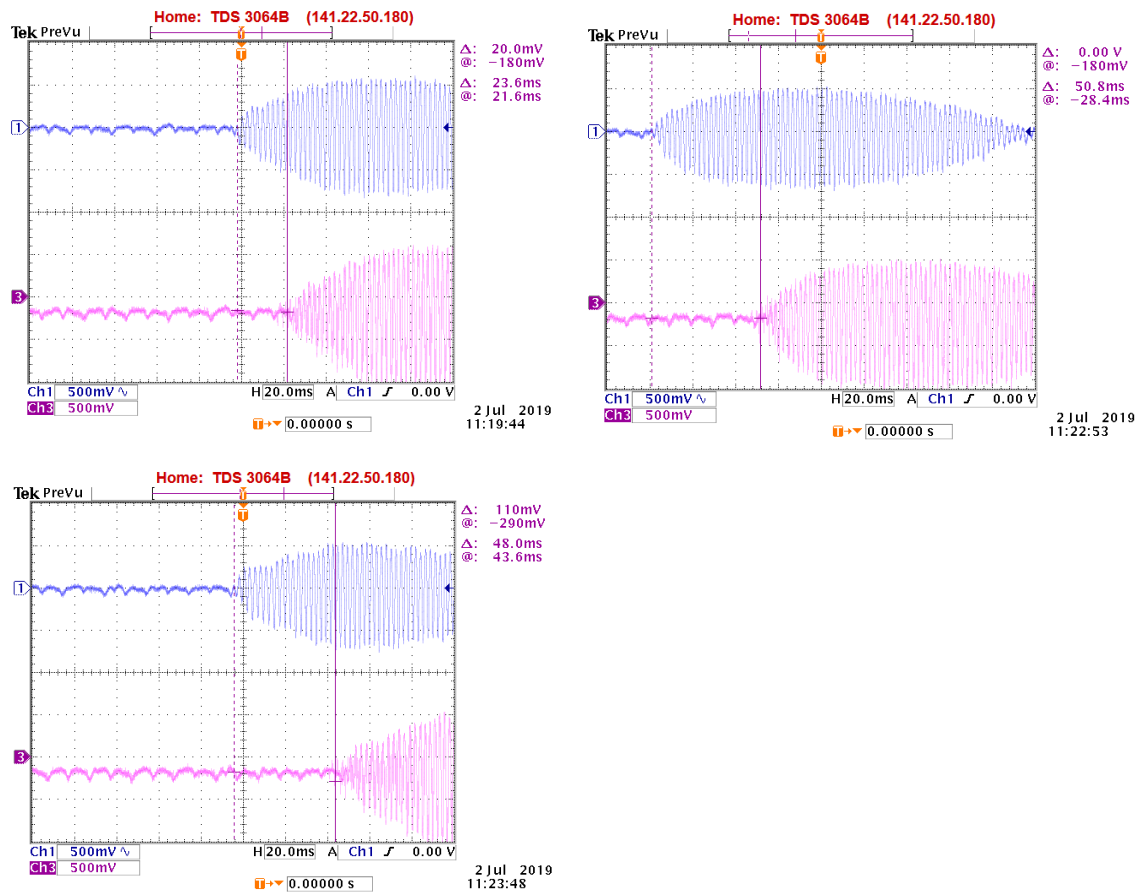


Abbildung 4.12: Messung der Latenz mit Oszilloskop für Schlüssel 256

## 4 Validierung

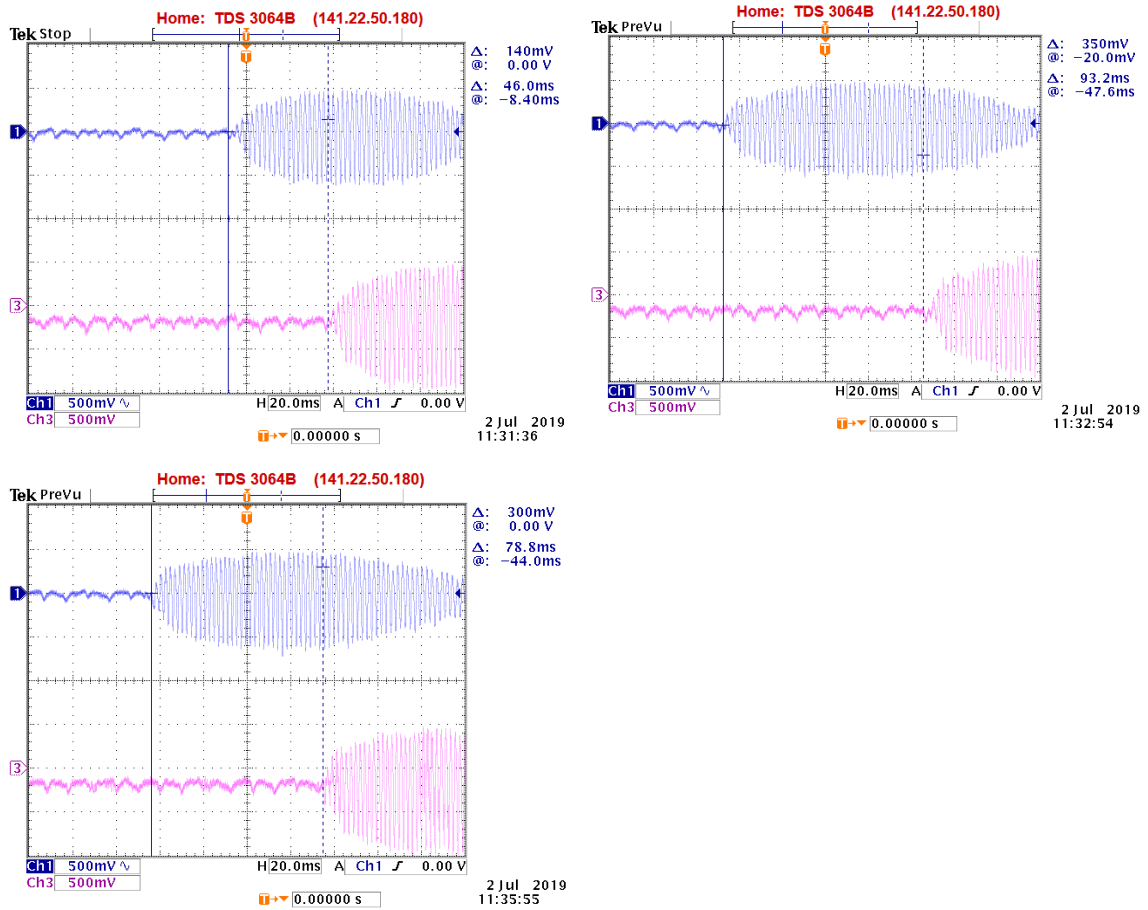


Abbildung 4.13: Messung der Latenz mit Oszilloskop für Schlüssel 512



## 4 Validierung

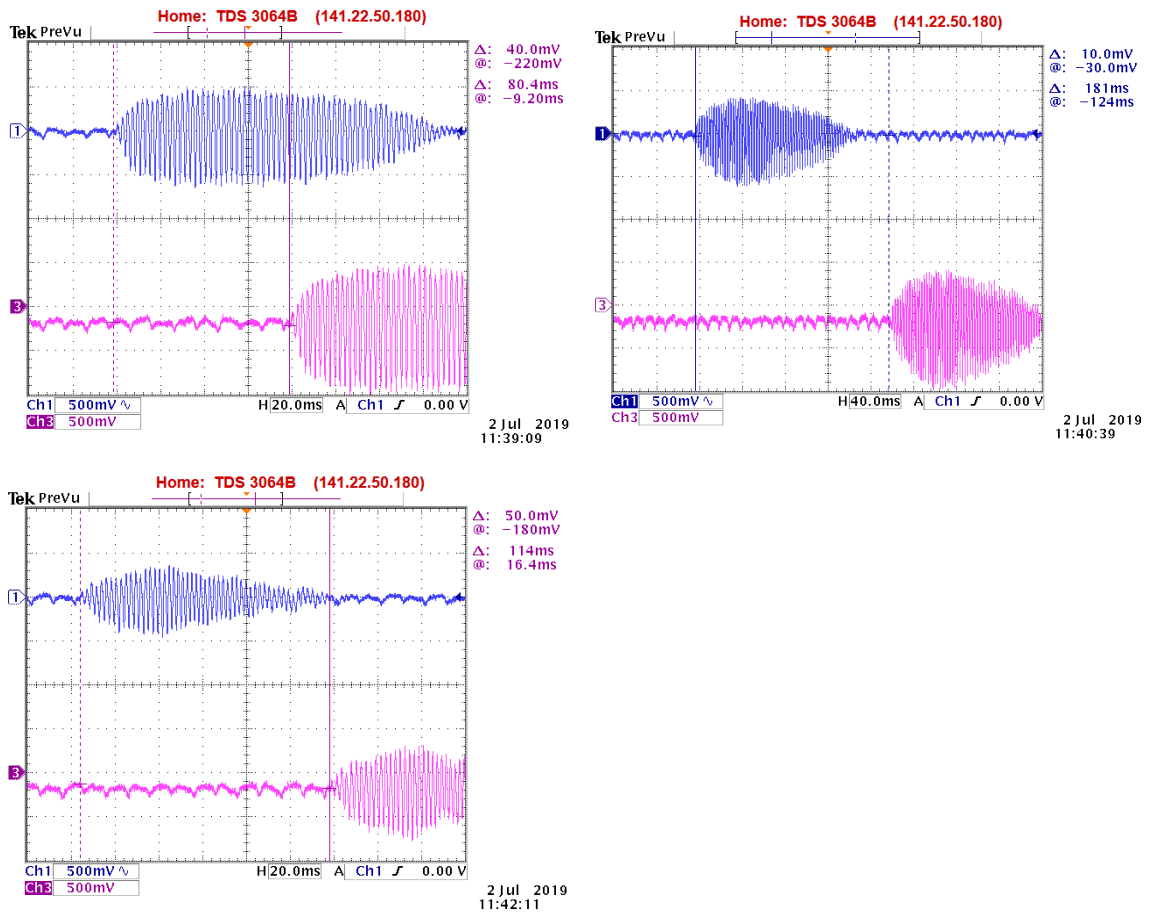


Abbildung 4.14: Messung der Latenz mit Oszilloskop für Schlüssel 1024

Gemessene Latenzen					
Schlüssel	D-Scope	M1	M2	M3	Mittelwert
128	30,5 ms	26,4 ms	30,8 ms	20,0 ms	26,93 ms
256	48,6 ms	23,6 ms	50,8 ms	48,0 ms	42,75 ms
512	87,03 ms	46,0 ms	93,2 ms	78,8 ms	76,26 ms
1024	152,7 ms	80,4 ms	181,0 ms	114,0 ms	132,03 ms

**Tabelle 4.1:** Gemessene Latenzen bei Verwendung unterschiedlicher Schlüssel (D-Scope: berechnet aus Phasenfrequenzgang, M1 bis M3: gemessen mit Oszilloskop)

Auch, wenn die einzelnen Werte nicht sehr genau sind, entspricht deren Mittelwert in etwa den erwarteten Werten. Die Verzögerung, die zusätzlich zu den nötigen Wandlungszeiten nötig sind, fallen bei kleineren Schlüssellängen mehr ins Gewicht. Auch das lässt sich beobachten. So konnte selbst für die Verwendung des Schlüssels 1024 eine Latenz gemessen werden, die unterhalb des gesetzten Ziels von 100 ms liegt.

### Interpretation der Ergebnisse

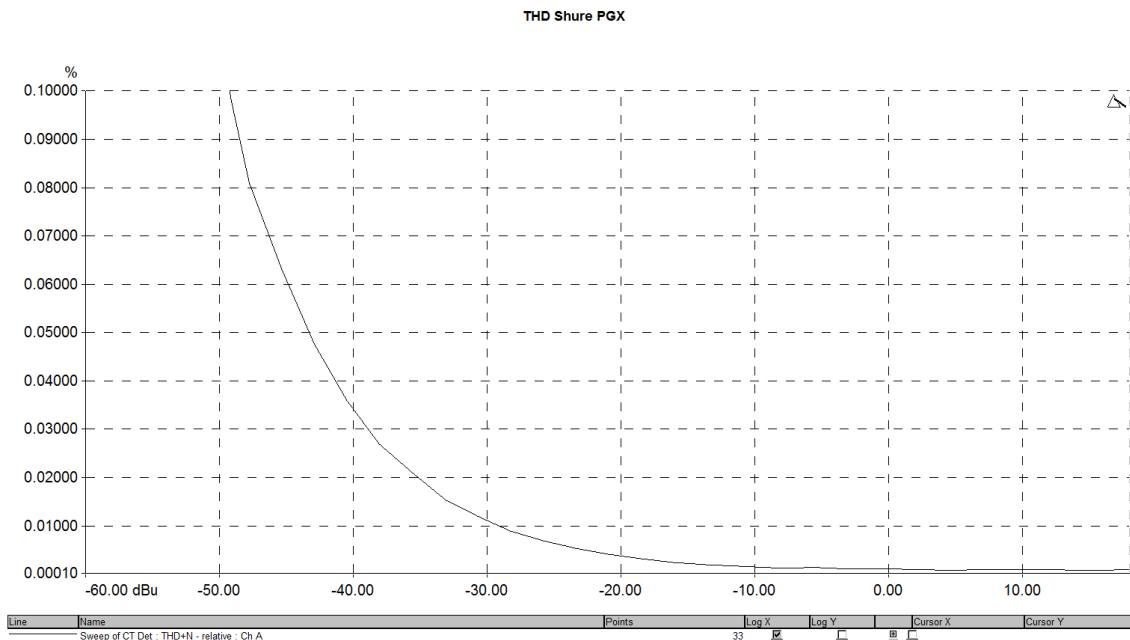
Es ist auffällig, dass teilweise Latenzen gemessen werden können, die unterhalb der theoretisch vorberechneten Werte (Siehe Tabelle 2.1) liegen. Dies lässt sich durch die Programmierung der Verschlüsselung erklären. Da die Werte des cryptBuffers bereits ausgegeben werden, bevor der Block komplett ver- oder entschlüsselt wurde, verschachteln sich nach einer sehr kurzen fehlerbehafteten Anfangszeit die einzelnen verschlüsselten Blöcke ineinander. Dadurch kann sich die Latenz des Gesamtsystems verkürzen, weil nicht auf die Ver- und Entschlüsselung eines zusammenhängenden Blockes gewartet wird. Außerdem auffällig ist allerdings, dass die Ergebnisse sehr stark schwanken und teilweise doppelt so hohe Werte wie erwartet, auch bei der Messung über das Oszilloskop, gemessen werden. Dies lässt sich zum Teil auch durch die Art der Programmierung erklären, da bei einer Störung nicht nur ein Block nicht korrekt entschlüsselt werden kann, sondern sich der Fehler für kurze Zeit fortpflanzt. Eine weitere Begründung könnte die Art der Messung sein. Die Messung der Latenz ist mit dem Oszilloskop nicht so genau, wie eine nach (DIN EN 61606-3 2009) durchgeführte Messung, wie sie im Idealfall mit dem D-Scope durchgeführt wird. Zur finalen Klärung der schwankenden Messergebnisse wäre es allerdings erforderlich, während der Messung zusätzlich die internen Daten des Chiffrierers und des Dechiffrierers auszuwerten. So könnte unter Umständen besser geprüft werden, wann der Phasenfrequenzgang linear verläuft und wann nicht und ob es eventuell Zusammenhänge mit nicht erkannten Barker-Codesequenzen gibt.

## 4.4 Klirrfaktor

Der Klirrfaktor gibt das Verhältnis des Effektivwerts der Oberwellen zum Effektivwert aller im Signal enthaltenen Wellen (Grundwelle und Oberwellen) in Prozent an ((Müller 2008: 1146)).

$$Klirrfaktor = 100\% \frac{\sqrt{\sum_{k=2}^n A_k^2}}{\sqrt{\sum_{k=1}^n A_k^2}} \quad (4.2)$$

Das Messsystem erfasst diese Oberschwingungen und stellt das Ergebnis graphisch dar. Der Klirrfaktor wird dabei in Abhängigkeit des Pegels bei einer Frequenz von 1 kHz gemessen, wobei der Pegel von -60 dBu bis +4 dBu in 1 dB-Schritten stetig erhöht wird ((Müller 2008: 1155)). Die Messung des Klirrfaktors des PGX-Systems



**Abbildung 4.15:** Shure PGX: THD+N

bestätigt die Angabe von einem Klirrfaktor unter 0,5 % im Datenblatt des Herstellers (Shure 2010). Die gemessenen Werte liegen sogar schon ab einem Pegel von -50 dBu bei 0,1% und damit deutlich unter der Herstellerangabe.

## 4 Validierung

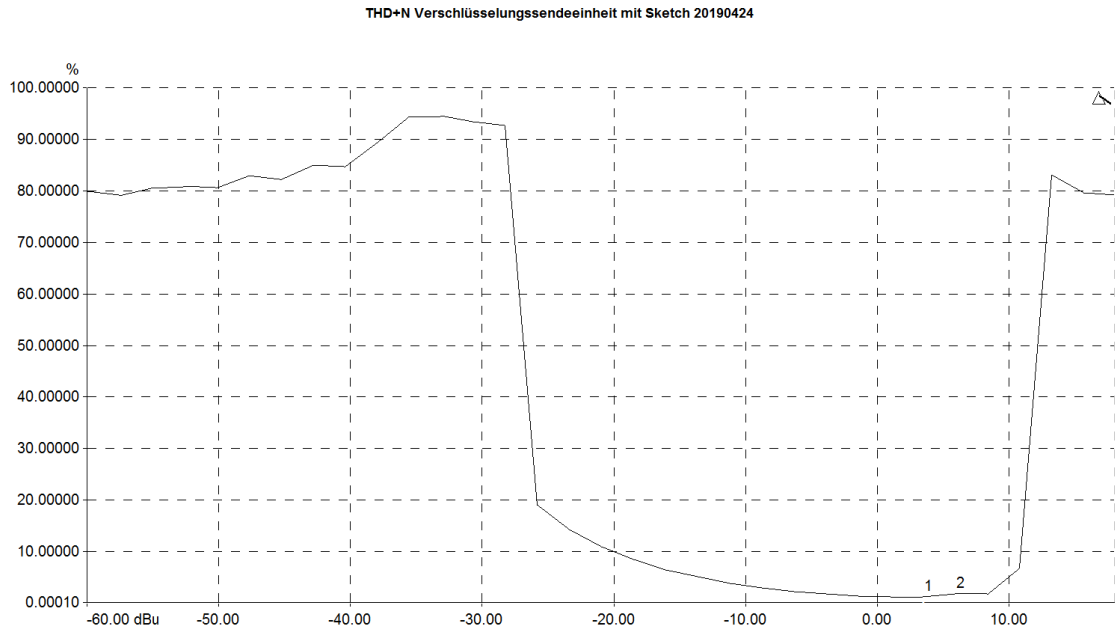


Abbildung 4.16: Chiffrierer: THD+N

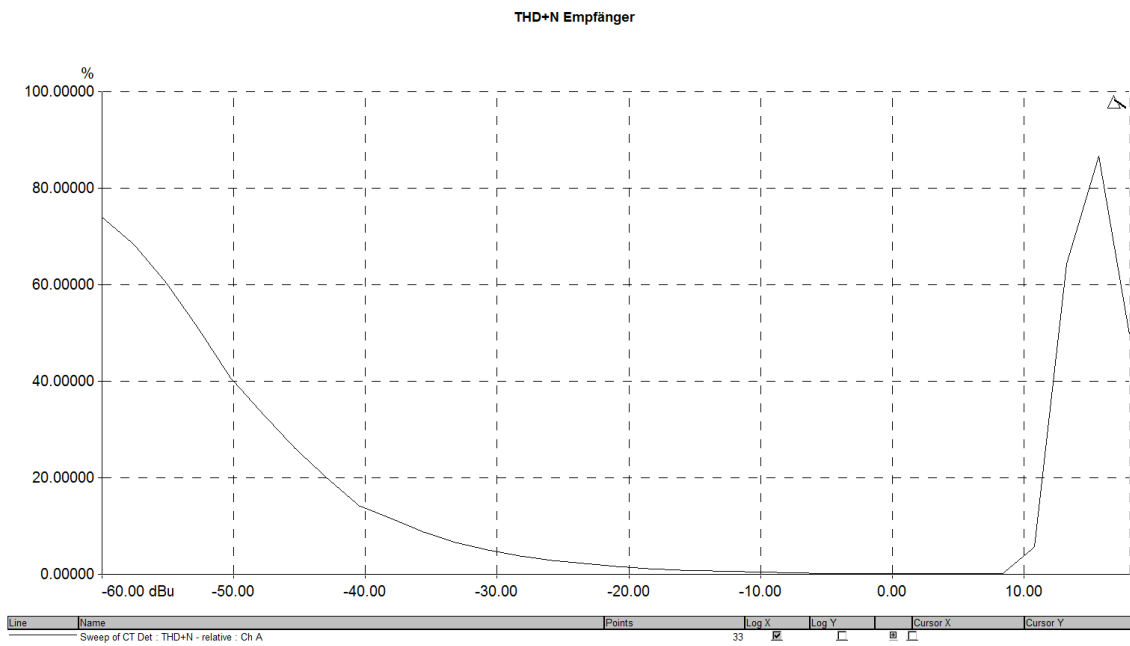


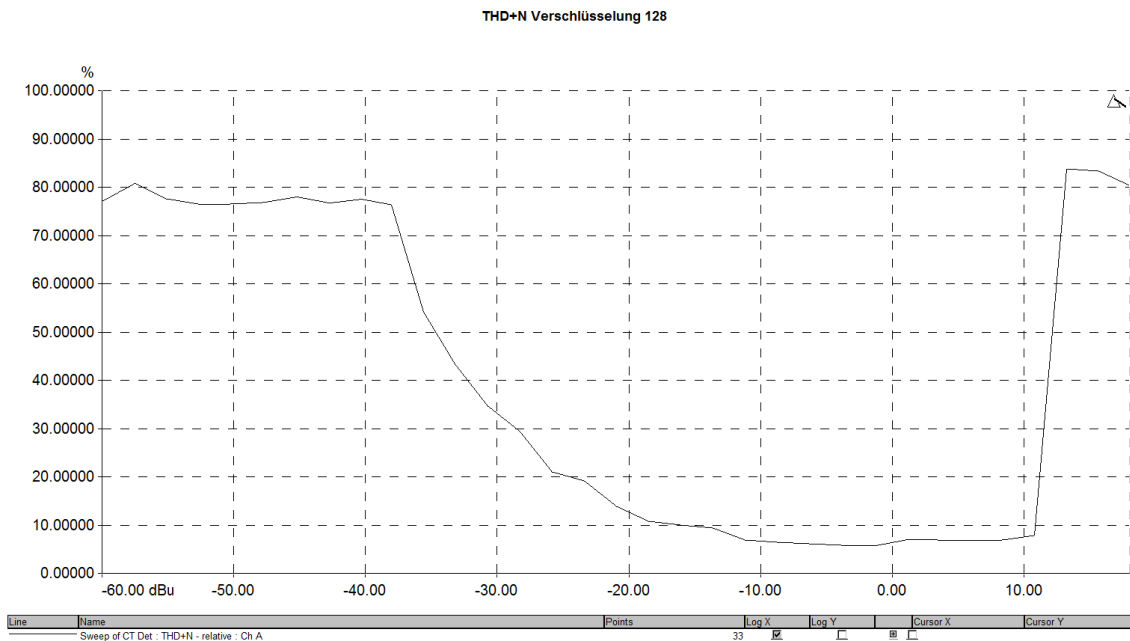
Abbildung 4.17: Dechiffrierer: THD+N

Wird der Klirrfaktor des Verschlüsselungssystems gemessen, sieht der Verlauf des

Graphen deutlich anders aus. Das liegt daran, dass es sich hierbei um ein digitales System handelt. Für geringe Pegel ist der Klirrfaktor sehr hoch, je nach Messung liegt er hier bei etwa 75% bis 90%. Das liegt daran, dass das Signal A/D-gewandelt wird. Ist hierfür das Signal nicht ausreichend ausgesteuert, fällt das Quantisierungsrauschen deutlich mehr ins Gewicht, da das Signal nur auf wenig verschiedene Quantisierungsstufen quantisiert wird. Deshalb ist der Sinus nach der D/A-Wandlung verformt, was in Oberschwingungen resultiert und so den hohen Klirrfaktor erklärt. Außerdem wird hier besonders die Qualität der Wandler sichtbar (Kapitel 3.4). Ab einem Pegel von etwa -30 dBu beträgt der Klirrfaktor jedoch weniger als 20 %. In einem Bereich von etwa -15 dBu bis +10 dBu liegt er deutlich unterhalb von 10%. Für höhere Pegel steigt der Klirrfaktor anschließend plötzlich an, da ab hier das Signal digital übersteuert wird.

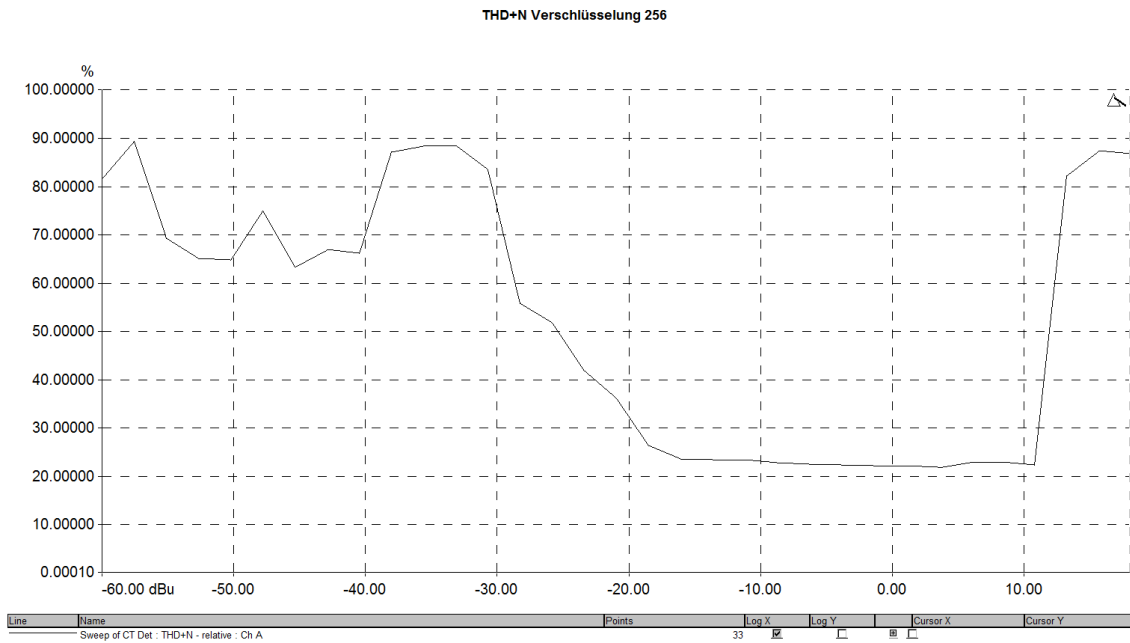
#### 4.4.1 Klirrfaktor bei unterschiedlichen Schlüsseln

Es ist zu erwarten, dass sich der Klirrfaktor durch die Verwendung unterschiedlicher Schlüssel nicht ändert.

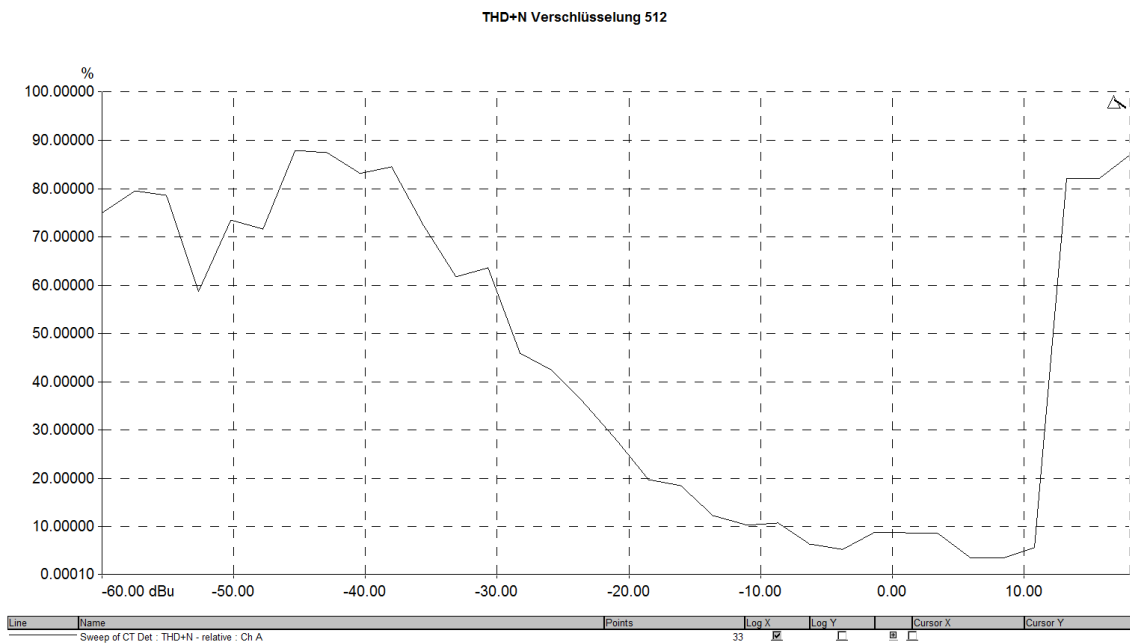


**Abbildung 4.18:** Verschlüsselung 128: THD+N

## 4 Validierung

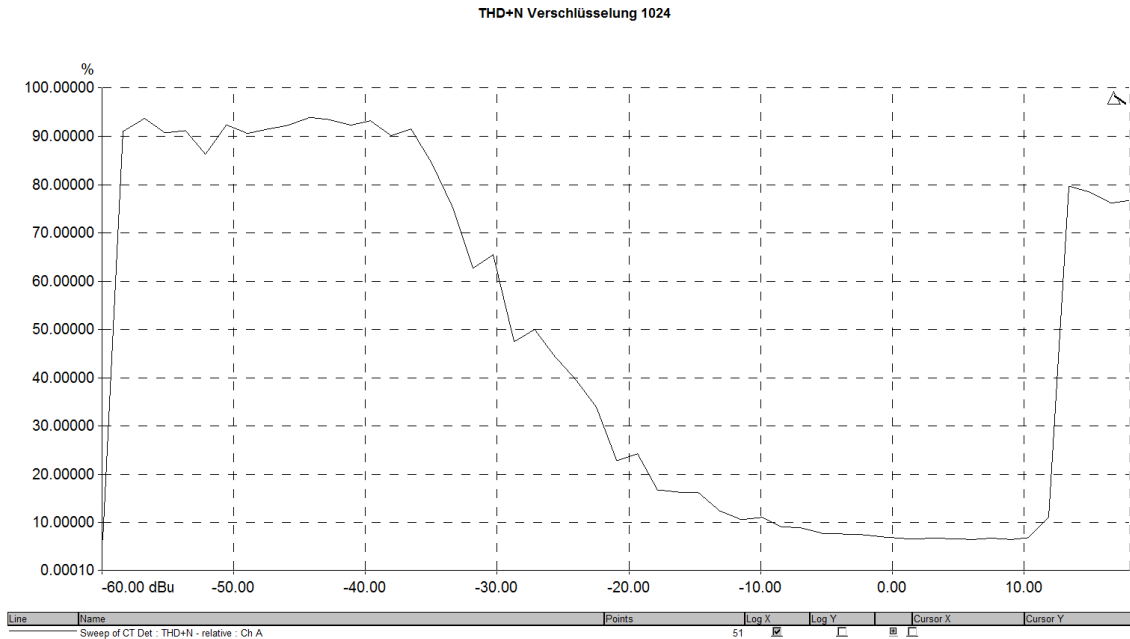


**Abbildung 4.19:** Verschlüsselung 256: THD+N



**Abbildung 4.20:** Verschlüsselung 512: THD+N

## 4 Validierung



**Abbildung 4.21:** Verschlüsselung 1024: THD+N

Die Messergebnisse bestätigen diese Erwartung grundsätzlich. Allerdings fällt der Graph für die Verwendung des Schlüssels 256 aus dem Rahmen (Abbildung 4.19). Hier liegt der niedrigste gemessene Wert bei über 20% und ist damit deutlich höher als bei der Verwendung der anderen Schlüssel. Bei allen anderen Schlüsseln liegt der niedrigste Wert bei etwa 8%. Die Abweichung kann erneut durch eine mögliche Störung während der Messung erklärt werden. Bei einer etwas versetzten Abtastung am Dechiffrierer im Vergleich zum Chiffrierer, werden Anteile der Barker-Codesequenz als Signalwerte gesehen und wie das Sinussignal entschlüsselt. Dies führt zu verteilten Signalspitzen, die dafür sorgen, dass Obertöne im Signal enthalten sind und der Klirrfaktor sich deshalb deutlich erhöht. Um dies aber als Grund bestätigen zu können, wäre es notwendig, die Messung zu wiederholen, um zu prüfen, ob auch ein Klirrfaktor mit einem Minimum unterhalb von 10% gemessen werden kann.

### 4.4.2 Bewertung der Ergebnisse

Ob der Klirrfaktor sich negativ auf die Signalqualität auswirkt, entscheidet sich dadurch, ob er für das menschliche Ohr hörbar ist. Dies ist wiederum sehr stark vom Eingangssignal abhängig. Da übertragene Signale anders als bei der Messung meist aus mehreren Sinusschwingungen bestehen, entstehen bei der Übertragung zusätzlich „Summen- und Differenzöne mit den Frequenzen  $f_1 + f_2$ ,  $f_1 - f_2$ ,  $2f_1 + f_2$ ,  $2f_1 - f_2$  etc.“ (Görne 2011: 230) Der Intermodulationsfaktor bezeichnet die so entstandenen nichtharmonischen Bestandteile des Signals im Verhältnis mit dem Effektivwert des

Gesamtsignal. Vor allem nichtharmonische Anteile werden als störend empfunden. Diese berücksichtigt der Klirrfaktor allerdings nicht. Trotzdem gibt er Aufschluss über die Qualität des Signals. Bei einem Klirrfaktor von etwa 8% ist davon auszugehen, dass der Mensch dies nicht zuletzt durch die ebenfalls auftretenden Intermodulationsverzerrungen hören kann.

### 4.5 Messung des Gesamtsystems mit Verschlüsselung

Interessant ist nun, wie das Verschlüsselungssystem im Zusammenspiel mit einer Mikrofonfunkstrecke funktioniert. Für die grundsätzliche Funktionalität ist entscheidend, ob die Barker-Codesequenz, obwohl deren Spektrum Frequenzen nahe der Übertragungsgrenzfrequenz der Mikrofonfunkstrecke enthält, korrekt übertragen wird und somit am Verschlüsselungsempfänger der Blockbeginn zuverlässig erkannt werden kann.

#### 4.5.1 Barker-Codesequenz über das System

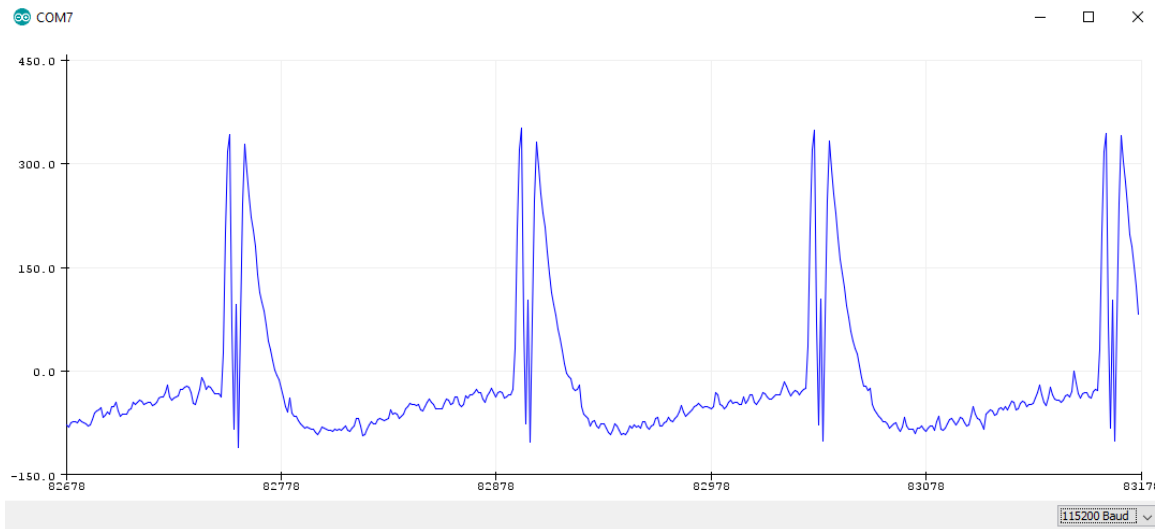
Um zu prüfen, ob dies funktioniert, ist es zunächst wichtig, sich die Werte der Barker-Codesequenz auf dem Verschlüsselungsempfänger anzeigen zu lassen. Gegebenenfalls muss dann der Schwellenwert neu angepasst werden. Für diesen Versuch wurde das System so aufgebaut, wie es bereits in der Planungsphase definiert wurde. Das angeschlossene Mikrofon, ein SM58 von Shure, hat jedoch für die ersten Messungen mit dem Ziel, die Synchronisierung zu prüfen, kein spezielles Signal aufgenommen, sondern lediglich den Raumklang übertragen.

Auf der Abbildung (4.23) ist gut zu erkennen, dass die Barker-Codesequenz periodisch empfangen wird. Allerdings entsprechen die empfangenen Werte nur noch entfernt den gesendeten. Außerdem ist zu erkennen, dass das Signal eine gewisse Zeit benötigt, um nach der Barker-Codesequenz wieder auf seinen ursprünglichen Wert zurückzukehren.

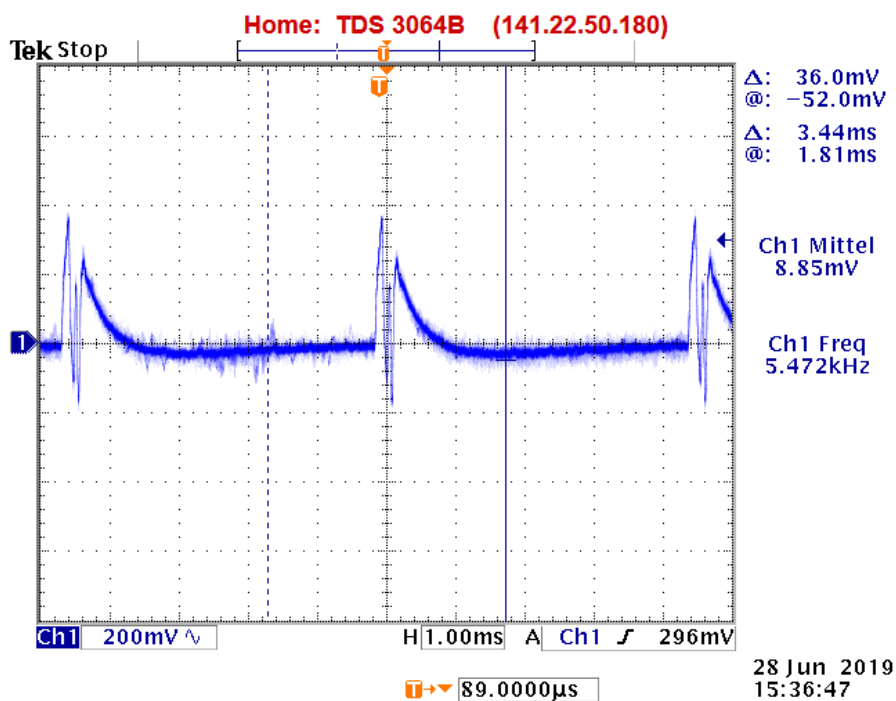
Die Dauer der Barker-Codesequenz beträgt  $344 \mu\text{s}$ . Da die eingestellte Abtastfrequenz bei  $30000 \text{ Hz} \cdot 17/16$  für Schlüssel 128 liegt und für die Sequenz acht Werte gesendet werden, wird eine Dauer von  $251 \mu\text{s}$  erwartet. Die Differenz zwischen erwartetem und gemessenem Wert führt also zu ähnlichen Problemen wie sie auch schon die in den Schaltungen verbauten Operationsverstärker verursachen. Teilweise sind diese hier mitverantwortlich, da das Signal am Output abgegriffen und zur Mikrofonfunkstrecke gesendet wird. Das größere Problem ist jedoch die Zeit, bis der ursprüngliche Pegel des eigentlichen Signals wieder erreicht wurde. Diese beträgt etwa eine Millisekunde, was knapp ein Viertel der gesamten Blocklänge bei einem Schlüssel von 128 ausmacht. Da nun diese deutlich anderen Werte durch den Dechiffrieralgorithmus wieder mit „korrekten“ Werten vertauscht werden, ist zu erwarten, dass das erhaltene Signal nicht oder nur schwer verständlich ist. Hinzu kommt die Frage, ob die Synchronisierung, die für die Entschlüsselung zwingend notwendig ist, mit den erhal-



## 4 Validierung



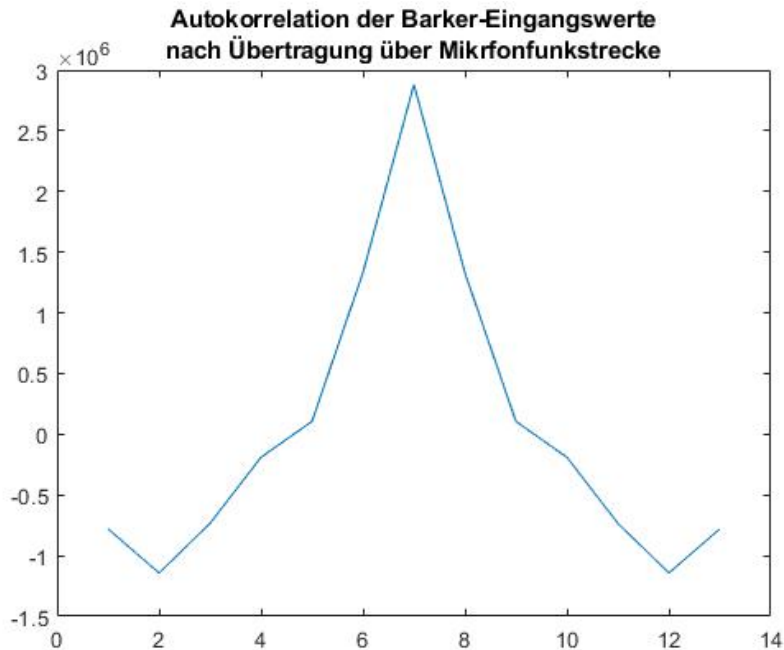
**Abbildung 4.22:** Eingangswerte am Verschlüsselungsempfänger bei Nutzung der Funkübertragungsstrecke



**Abbildung 4.23:** Spannung am Ausgang des Funkempfängers PGX4 von Shure

## 4 Validierung

tenen Werten funktionieren kann. Führt man die Autokorrelation dieser Werte mit Hilfe von Matlab durch, erhält man das folgende Ergebnis (Abbildung 4.24):



**Abbildung 4.24:** Autokorrelation der gemessenen Werte der Barker-Codesequenz nach der Übertragung über die Mikrofonfunkstrecke

Es ist zu erkennen, dass das Signal immer noch ein Maximum besitzt, welches aber deutlich schwächer ausgeprägt ist als es für das Autokorrelationsergebnis der Barker-Codesequenz der Länge sieben möglich wäre. Der Maximalwert liegt bei etwa 29000000, somit wird ein Wert unterhalb dieses Wertes als Schwellenwert im Sketch eingestellt. Außerdem wird die dort gespeicherte Barker-Codesequenz geändert, so dass das Eingangssignal mit der gemessenen Barker-Codefolge gefaltet wird. Nun ist es möglich, das Signal zu synchronisieren, der Vorgang läuft allerdings nicht stabil und produziert ab und zu Fehler. Das passiert, wenn beispielsweise eine Eingangssequenz, die nicht die Barker-Codesequenz ist, zu einem Faltungsergebnis über dem Schwellenwert führt. Dann werden die falschen Werte als Block interpretiert und entschlüsselt, was wiederum dazu führt, dass dieser Block als kurze Störung zu hören ist. Der andere Fall ist, dass das Faltungsergebnis unterhalb des Schwellenwertes liegt, obwohl eigentlich ein Blockbeginn erkannt werden sollte. Das führt dazu, dass unter Umständen keine Werte am im Ausgangsspeicher zur Verfügung stehen und dadurch auch nichts gesendet wird. Eine weitere Beobachtung ist der sehr geringe Pegel der Barker-Codesequenz. Dieser lässt sich durch seine hohe Frequenz erklären, die sowieso schon durch die analoge Schaltung des Verschlüsselungssenders stark bedämpft ist. Sie entspricht ebenfalls nahezu der vom Hersteller angegebenen Übertragungsgrenz-

frequenz von 15 kHz (Shure 2010). Daher ist davon auszugehen, dass dies auch dazu beiträgt, dass der Pegel der Barker-Folge so stark fällt. Wird ins Mikrofon gesprochen und nicht nur der Raumklang aufgenommen, ist am Oszilloskop bei Messung der Spannung am Ausgang des Funkempfängers zu beobachten, dass die Blöcke des Sprachsignals mit einem deutlich höheren Pegel empfangen werden als die Barker-Codesequenz. Dies erschwert die Synchronisation zusätzlich.

### 4.5.2 Bewertung der Ergebnisse

Zusammenfassend lässt sich also festhalten, dass das Verschlüsselungssystem in Kombination mit der verwendeten Mikrofonfunkstrecke nur sehr eingeschränkt funktioniert und kein verständliches Audiosignal liefert. Interessant wäre nun zu wissen, warum das Signal nach Senden der Barker-Codesequenz so viel Zeit benötigt, bis es wieder auf seinem ursprünglichen Pegel angekommen ist und weshalb der Pegel dieser Signalabschnitte am Funkempfänger so viel geringer als am Funksender ist.

Einen Einfluss auf den geringen Pegel der Barker-Codesequenz könnte der im PGX-System verwendete Kompander (Shure 2010) haben.

Ein Kompander dient dazu, den Signalrauschabstand der Funkübertragungsstrecke zu erhöhen. Dafür wird das Signal auf Senderseite komprimiert und auf Empfängerseite wieder expandiert. Durch die Komprimierung kann das Signal nun vom Sender höher ausgesteuert werden. Das Rauschen ist allerdings bei der Komprimierung und anschließenden Pegelanhebung noch nicht Teil des Signals. Am Funkempfänger wird das gesamte Signal wieder expandiert, also in seinem Dynamikumfang vergrößert. So wird auch das Signal inklusive dem hinzugekommenen Rauschen, das zuvor nicht angehoben wurde, wieder auf seinen ursprünglichen Pegel absenkt (Niehoff 2008: 1043). Die Frage ist nun, wie genau die neue Aussteuerung des Signals nach der Komprimierung am Sender funktioniert. Als Nutzer gibt es keine Möglichkeit, diese zu beeinflussen, sondern sie geschieht automatisch. Die Barker-Codesequenz ist der Teil des Signals, der durch die Wahl der minimal und maximal möglichen Abtastwerte maximal ausgesteuert wird. Eigentlich wäre allerdings zu erwarten, dass der Expander im Funkempfänger so eingestellt ist, dass er die Kompression prinzipiell wieder rückgängig macht. Allerdings gibt der Hersteller an, der verwendete Kompandersystem, die patentierte „Audio Reference Companding“ Technologie“ (Shure 2010) selbst entwickelt hat, dass dieser ein „pegelabhängiges Kompressionsverhältnis“ (Shure 2010) besitzt. Dies führt dazu, dass die Barker-Codesequenz, die immer das am höchsten ausgesteuerte Signal ist, mit dem höchsten Kompressionsverhältnis komprimiert wird. Das erklärt allerdings noch nicht, weshalb diese Sequenz im Vergleich zu einem Audiosignal sogar niedriger ausgesteuert am Empfänger gemessen werden kann. Daher ist unklar, ob dieses System neben der Erhöhung des Signalrauschabstands durch die Kompandertechnologie noch weitere Aufgaben erfüllt und beispiels-

weise durch einen Limiter (<sup>4</sup> die verhältnismäßig kurzen, aber hoch ausgesteuerten Barker-Codesequenzen zusätzlich weiter heruntergeregelt werden. Um das allerdings beurteilen zu können, müsste genauer bekannt sein, wie die Mikrofonfunkstrecke und besonders das Kompendersystem im Detail aufgebaut ist.

## 4.6 Hörtest zur Validierung der Verschlüsselung

### 4.6.1 Aufbau und Zielsetzung

Das Ziel des Hörtests ist es herauszufinden, ob das zu Beginn gesetzte Ziel der Sprachunverständlichkeit beim Mithören des verschlüsselten Signals erreicht wurde. Zusätzlich ist interessant, ob und welche Informationen ein unbefugter Mithörer aus dem verschlüsselten Signal ziehen kann. Dazu zählt beispielsweise die Information, ob es sich um eine weibliche oder männliche Stimme handelt. Für den Test werden außerdem Sprachsignale, die mit unterschiedlichen Schlüsseln verschlüsselt wurden, verwendet. So soll herausgefunden werden, ob die Wahl des Schlüssels eine Auswirkung auf die Verständlichkeit hat. Außerdem soll untersucht werden, ob Sätze, die sich bereits unverschlüsselt ähneln, fälschlicherweise für den selben Satz gehalten werden oder ob sie auch im verschlüsselten Zustand unterschieden werden können. Generell geht es dabei aber nicht darum, dass der Hörer das Signal speichern und anschließend beliebig oft abspielen und analysieren kann, sondern lediglich um das einmalige Mithören.

Hierfür werden dem/der Testhörer(in) insgesamt sechs ausgewählte Audiodateien vorgespielt. Der/die Hörer(in) beantwortet jeweils nach einmaligem Anhören der einzelnen Dateien die zugehörigen Fragen. Durch das Verwenden von aufgenommenen Audiodateien, erfolgt eine bessere Vergleichbarkeit.

Im Anschluss folgt eine Diskussion mit dem/der Testhörer(in), bei der ausgewählte Dateien noch einmal beliebig oft angehört werden können, um so festzustellen, ob sich der erste Eindruck der Testperson dadurch wieder verändert. Außerdem soll im Gespräch herausgefunden werden, ob die Testperson erkennen kann, ob oder welche Audiodateien identisch waren und woran sie dies festmachen würde. Für den Hörtest wurden folgende Audiodateien ausgewählt:

- A: M3-128-Bark
- B: F3-128-Matlab
- C: F1-512-D
- D: M2-1024-Bark
- E: M1-1024-Bark
- F: F1-256-E

---

<sup>4</sup>Ein Limiter ist ein Kompressor mit einem Kompressionsverhältnis ab etwa 1:10 bis 1:20 (Görne 2011: 359)

Es werden drei verschiedene Sätze verwendet, die jeweils von einem männlichen Sprecher und einer weiblichen Sprecherin eingesprochen wurden. Erst anschließend werden diese Dateien mit Hilfe des Verschlüsselungssystems unter Verwendung verschiedener Schlüssel verschlüsselt und wieder aufgezeichnet. Bei Datei A handelt es sich um einen männlichen Sprecher, der den Satz „Dies ist ein Test, ob das verschlüsselte Sprachsignal verständlich ist.“ spricht. Datei B ist derselbe Satz mit dem selben Schlüssel, jedoch von einer weiblichen Person eingesprochen und nicht über das Verschlüsselungssystem, sondern über das Programm Matlab (Skript siehe Kapitel 2.2.3) verschlüsselt. Bei den Dateien C, E und F handelt es sich um den Satz „Willkommen zur Präsentation.“, Datei D ist der sehr ähnlich klingende Satz „Wir kommen zur S-Bahn-Station.“. Da sich die Sprachstruktur im Zeitbereich kaum vom unverschlüsselten Signal unterscheidet, soll so überprüft werden, ob ihre verschlüsselten Versionen voneinander unterschieden werden können. Deshalb handelt es sich bei den Dateien D und E auch jeweils um einen männlichen Sprecher und den Schlüssel 1024. Datei C ist der mit Schlüssel 512 ver- und wieder entschlüsselte Satz, jedoch von einer Frau gesprochen. Auch in Datei F handelt es sich um eine weibliche Stimme, jedoch wurde hier die Synchronisierungssequenz nicht durch den Verschlüsselungsempfänger entfernt. Dies ist jedoch das einzige Signal, das dem entspricht, was ein unbefugter Mithörer tatsächlich empfangen würde. Allerdings ist die Synchronisierungssequenz nur indirekt Teil der Verschlüsselung, da das Verfahren allgemein bekannt ist und davon ausgegangen werden muss, dass der unbefugte Mithörer in der Lage wäre, dieses zu entfernen. Trotzdem ist es interessant, ob auch bei diesem Signal die Sprachstruktur erkannt werden kann.

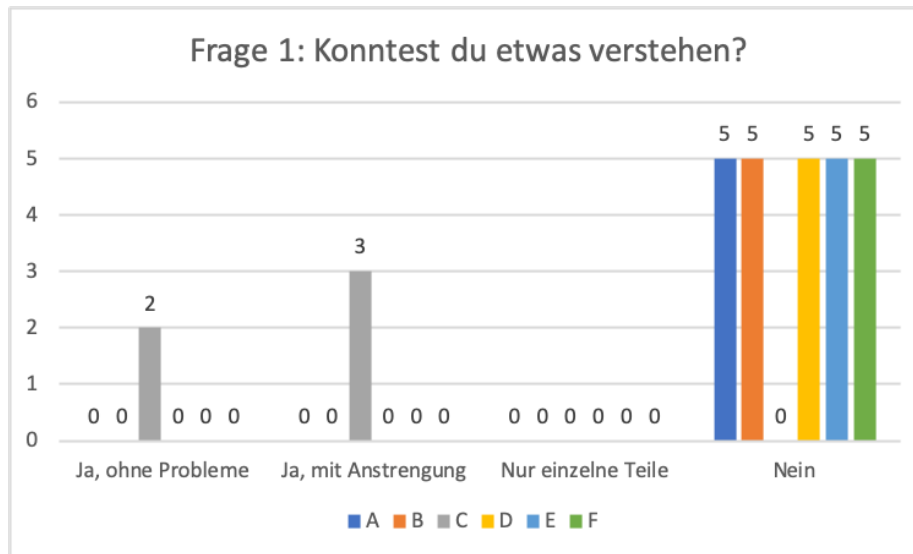
### 4.6.2 Ergebnisse und Bewertung

Die Teilnehmerzahl des Hörtests liegt bei fünf Personen, daher sind die Ergebnisse nicht repräsentativ. Allerdings lässt sich trotz der geringen Teilnehmerzahl eine klare Tendenz der Ergebnisse erkennen.

Erwartet wurde, dass Datei C verständlich ist und das Geschlecht richtig zugeordnet werden kann.

Vergleicht man nun die Ergebnisse der ersten Frage (Abbildung 4.25), lässt sich schnell erkennen, dass alle verschlüsselten Dateien von keiner Testperson verstanden werden konnten. Die entschlüsselte Datei C wurde hingegen von allen verstanden, jedoch gaben drei Personen an, sie nur mit Anstrengung verstehen zu können. Die Eintragung des Satzes bei Frage 3 ist aber bei allen Personen korrekt.

Alle Personen haben ebenfalls erkannt, dass es sich in Datei C um eine weibliche Stimme handelt. Interessanter sind jedoch die Ergebnisse für A und B. Datei A haben drei von fünf Personen richtig als männliche Stimme gehört, eine vierte Person, tendierte zur richtigen Angabe, kreuzte aber dennoch das Feld „Weiß ich nicht“ an. Bei Datei B erkannte eine Person die Stimme richtig als weiblich. Da bei beiden Dateien der Schlüssel 128 verwendet wurde, liegt die Vermutung nahe, dass bei klein gewählten Blocklängen das Geschlecht des Sprechers noch erahnt werden kann. Jedoch müsste,



**Abbildung 4.25:** Übersicht der Ergebnisse des Hörtests für Frage 1

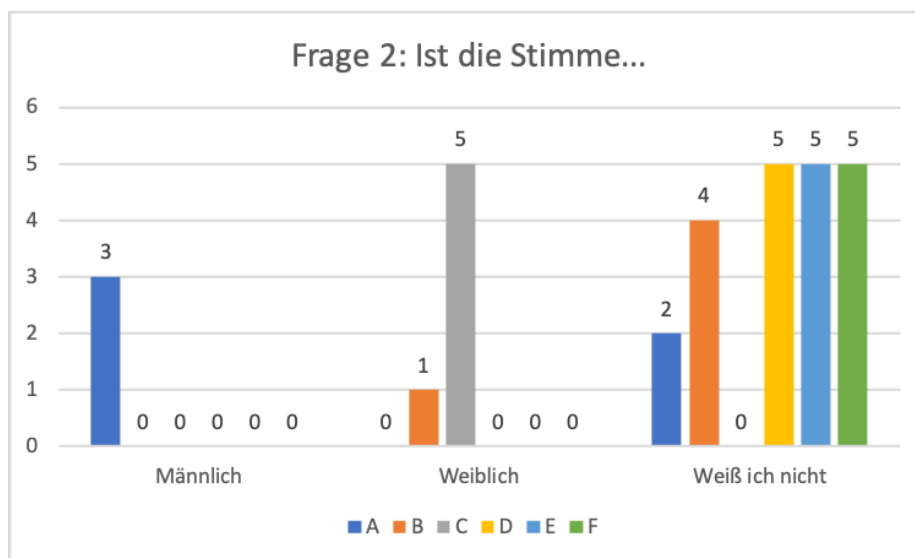
um diese Aussage treffen zu können ein weiterer Test mit deutlich mehr Testpersonen durchgeführt werden, der sich speziell auf diesen Aspekt fokussiert. Hierfür wäre es ebenfalls wichtig, mehr Testpersonen zu befragen, um diese Vermutung entweder zu bestätigen oder zu widerlegen. Für die Dateien D, E und F konnten alle Testpersonen keine Geschlechtszuordnung machen.

Im zweiten Teil der Befragung waren die Ergebnisse unterschiedlich (Abbildung 4.26). Zwei der befragten Personen vermuteten, dass es sich bei den Dateien A und B um dasselbe Signal handelte. Eine weitere Person war der Meinung, alle Dateien seien identisch gewesen. Die Dateien D und E wurden von insgesamt drei Personen als identisch eingestuft, eine weitere Person vermutete, dass auch Datei F dasselbe sei. Wenn nach den Kriterien, an denen die Teilnehmer diese Aussagen festmachten, gefragt wurde, gaben allerdings alle an, sich ausschließlich an der Struktur, der Länge und dem Rhythmus orientiert zu haben. Ein Teilnehmer gab an, dass es leicht zu erkennen sei, dass es sich bei den Signalen um Sprache handle. Alle waren sich jedoch einig, dass sie auch nach mehrmaligem Hören nichts verstehen konnten. Eine Person konnte jedoch mit dem Wissen über den Inhalt der Datei B diese bei mehrmaligem Hören teilweise verstehen.

Auch hier gilt wieder, dass diese Ergebnisse keinesfalls repräsentativ sind. Dennoch legen sie die Vermutung nahe, dass die schon unverschlüsselt sehr ähnlich klingenden Sätze aus Datei D und E auch bei einer Umfrage mit mehr Testpersonen nicht unterschieden werden könnten.

Insgesamt lässt sich allerdings festhalten, dass das Ziel der Unverständlichkeit erreicht wurde.

## 4 Validierung



**Abbildung 4.26:** Übersicht der Ergebnisse des Hörtests für Frage 2

## 5 Zusammenfassung

Um analoge Mikrofonfunkstrecken auch in einer abhörsicheren Umgebung nutzen zu können, erfordert dies eine Verschlüsselung. Die Frage, inwieweit eine analoge Mikrofonfunkstrecke durch ein Verschlüsselungssystem ergänzt werden kann und so auch bei der Nutzung analoger Funkstrecken eine abhörsichere Umgebung sichergestellt werden kann, wurde in dieser Arbeit beantwortet.

Es eignen sich hierfür Verschlüsselungsverfahren, die das Signal mittels Transposition verschlüsseln. Dies erfordert eine Rahmensynchronisierung, um die jeweiligen Blockanfänge zu kennen, für die die Barker-Codesequenz verwendet wurde. Die Programmierung der Rahmenerkennung könnte allerdings noch verbessert werden, indem eine echte Autokorrelation durchgeführt wird und sich der Schwellenwert dynamisch an den Kanal anpasst.

Es ist theoretisch mit dem entworfenen und entwickelten System möglich, Audiosignale verschlüsselt in Echtzeit zu übertragen, ohne dass bei Abhören das Signal verstanden werden kann. Dies wird durch die Ergebnisse des Hörtests belegt. Auch die Echtzeitnutzung, die ebenfalls zwingend notwendig ist, damit das Gesamtsystem nutzbar wird, ist je nach verwendetem Schlüssel gegeben.

Vor allem aber die Zwischenschaltung einer Mikrofonfunkstrecke, für die das System eigentlich vorgesehen ist, sorgt für Probleme. Es lässt sich festhalten, dass das umgesetzte Verschlüsselungssystem mit der verwendeten Mikrofonfunkstrecke nicht nutzbar ist. Durch die Umsetzung einer digitalen Chiffrierung und die damit verbundenen nötigen A/D- und D/A-Wandlungen treten Störsignale auf, die bei der Übertragung über die Mikrofonfunkstrecke nicht kontrollierbar sind. Es fehlt außerdem eine gemeinsame Taktung der Sender- und Empfängerseite, die durch eine und Leitungscodierung erreicht werden könnte.

Es wäre interessant zu untersuchen, ob das System mit anderen analogen Mikrofonfunkstrecken eventuell besser funktioniert und welchen Einfluss das Compandersystem auf das verschlüsselte Signal und die Synchronisierungssequenz hat.

In der Arbeit wurden einige Ideen zu möglichen Verbesserungen vorgestellt, damit das System einfacher zu nutzen und sicherer wird. Dazu zählen die Vereinfachung des Schlüsselaustauschs und die dynamisch angepasste Rahmenerkennung (siehe oben). Außerdem kann die umgesetzte analoge Schaltung so verbessert werden, dass Störungen, die durch verzögernde Operationsverstärker zu erklären sind, nicht mehr auftreten. Durch die Umsetzung Tiefpassfilter höherer Ordnung könnte außerdem die Audioqualität verbessert werden und der Frequenzgang linearisiert werden. Vor allem aber müsste zunächst die Funktionalität in Kombination mit einer Mikrofonfunkstrecke erreicht werden. Daher wäre es interessant, genauer zu untersuchen, warum diese



## 5 Zusammenfassung

dafür sorgt, dass das gesendete Signal qualitativ nicht mehr ausreichend ist, um es tatsächlich gemeinsam zu nutzen.

Zusätzlich wäre für eine Weiterentwicklung die Verwendung eines DSP mit hochqualitativeren Wandlern anstelle des Arduino Due mit den dort verbauten ADCs und DACs interessant.

Es wäre insgesamt begrüßenswert, wenn die vorgeschlagenen Verbesserungsmöglichkeiten umgesetzt würden und noch weiter untersucht werden würde, wo die Grenzen eines Echtzeitverschlüsselungssystems für Mikrofonfunkstrecken liegen. Die langfristige tatsächliche professionelle Umsetzung eines solchen Systems ist allerdings unwahrscheinlich, da es bereits deutlich sicher verschlüsselndere digital übertragende Systeme auf dem Markt gibt.

# A Material

## A.1 Beigefügte CD

Auf der beigefügten CD befinden sich folgende Inhalte:

- Die vorliegende Arbeit als pdf
- Der Ordner „Audiodateien“, der alle erwähnten Audiodateien der Liste A.2 enthält
- Der Ordner „Matlab“, der die verwendeten Matlab-Skripte sowie eine Anleitung zur Nutzung als pdf enthält
- Der Ordner „Arduino“, der alle verwendeten Arduino-Sketches sowie eine Anleitung zur Nutzung als pdf enthält

## A.2 Liste über die Benennung und Inhalt der Audiodateien

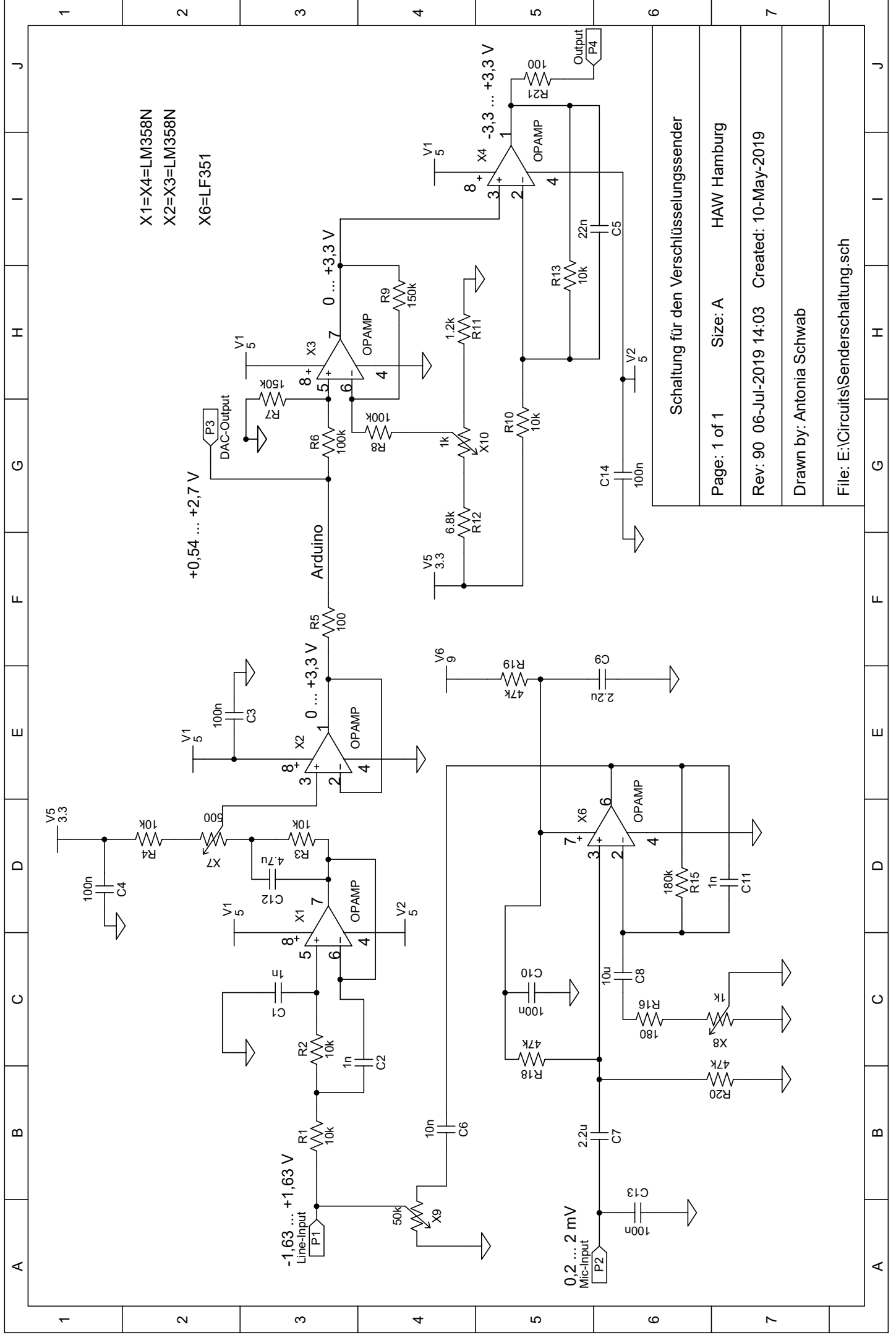
Folgende Audiodateien befinden sich im Ordner „Audiodateien“ auf der beigefügten CD:

- M1-Original: Originalsignal „Willkommen zur Präsentation.“, männliche Stimme
- M2-Original: Originalsignal „Wir kommen zur S-Bahn-Station.“, männliche Stimme
- M3-Original: Originalsignal „Dies ist ein Test, ob das verschlüsselte Sprachsignal verständlich ist.“, männliche Stimme
- F1-Original: Originalsignal „Willkommen zur Präsentation.“, weibliche Stimme
- F2-Original: Originalsignal „Wir kommen zur S-Bahn-Station.“, weibliche Stimme
- F3-Original: Originalsignal „Dies ist ein Test, ob das verschlüsselte Sprachsignal verständlich ist.“, weibliche Stimme
- F3-OTP65536: F3-Original mit One Time Pad mit Schlüssel 65536 verschlüsselt

- F3-OTP65536-K: Entschlüsseltes Signal F3-Original mit One Time Pad mit Schlüssel 65536 verschlüsselt und über simulierten Kanal mit 50 dB SNR und einer Dämpfung mit Faktor 0,9 geschickt
- F3-OTP65536-D: Entschlüsseltes Signal F3-Original mit One Time Pad mit Schlüssel 65536 ver- und entschlüsselt ohne Kanalübertragung
- F3-4-Matlab: F3-Original mit Blocklänge 4 ohne Verschachtelung verschlüsselt
- F3-128-rückwärts: F3-Original mit Blocklänge 128 ohne Verschachtelung verschlüsselt
- F3-128-Matlab: F3-Original mit Blocklänge 128 mit Verschachtelung verschlüsselt
- F3-128-Matlab-K1: F3-128-Matlab über Kanal übertragen (Dämpfung 0,9, SNR 80dB)
- F3-128-Matlab-K2: F3-128-Matlab über Kanal übertragen (Dämpfung 0,9, SNR 20dB)
- F3-256-Matlab: F3-Original mit Blocklänge 256 verschlüsselt
- F3-512-Matlab: F3-Original mit Blocklänge 512 verschlüsselt
- F3-1024-Matlab: F3-Original mit Blocklänge 1024 verschlüsselt
- M3-128-Bark: M3-Original mit Chiffrierer mit Schlüssel 128 verschlüsselt, mit Dechiffrierer nur Barker-Codesequenz entfernt, aber nicht entschlüsselt
- F1-512-D: F1-Original mit Chiffrierer mit Schlüssel 512 verschlüsselt, mit Dechiffrierer entschlüsselt
- M2-1024-Bark: M2-Original mit Chiffrierer mit Schlüssel 128 verschlüsselt, mit Dechiffrierer nur Barker-Codesequenz entfernt, aber nicht entschlüsselt
- M1-1024-Bark: M1-Original mit Chiffrierer mit Schlüssel 128 verschlüsselt, mit Dechiffrierer nur Barker-Codesequenz entfernt, aber nicht entschlüsselt
- F1-256-E: F1-Original mit Chiffrierer mit Schlüssel 256 verschlüsselt inklusive Barker-Codesequenz

„F“ kennzeichnet dabei eine weibliche, „M“ eine männliche Stimme. Die 1 steht immer für den Satz „Willkommen zur Präsentation.“, die 2 für den Satz „Wir kommen zur S-Bahn-Station.“ und die 3 für den Satz „Dies ist ein Test, ob das verschlüsselte Sprachsignal verständlich ist.“.

### **A.3 Analoge Schaltungen des Chiffrierers und Dechiffrierers**



X1=X4=LM358N  
 X2=X3=LM358N  
 X6=LF351

Schaltung für den Verschlüsselungssender

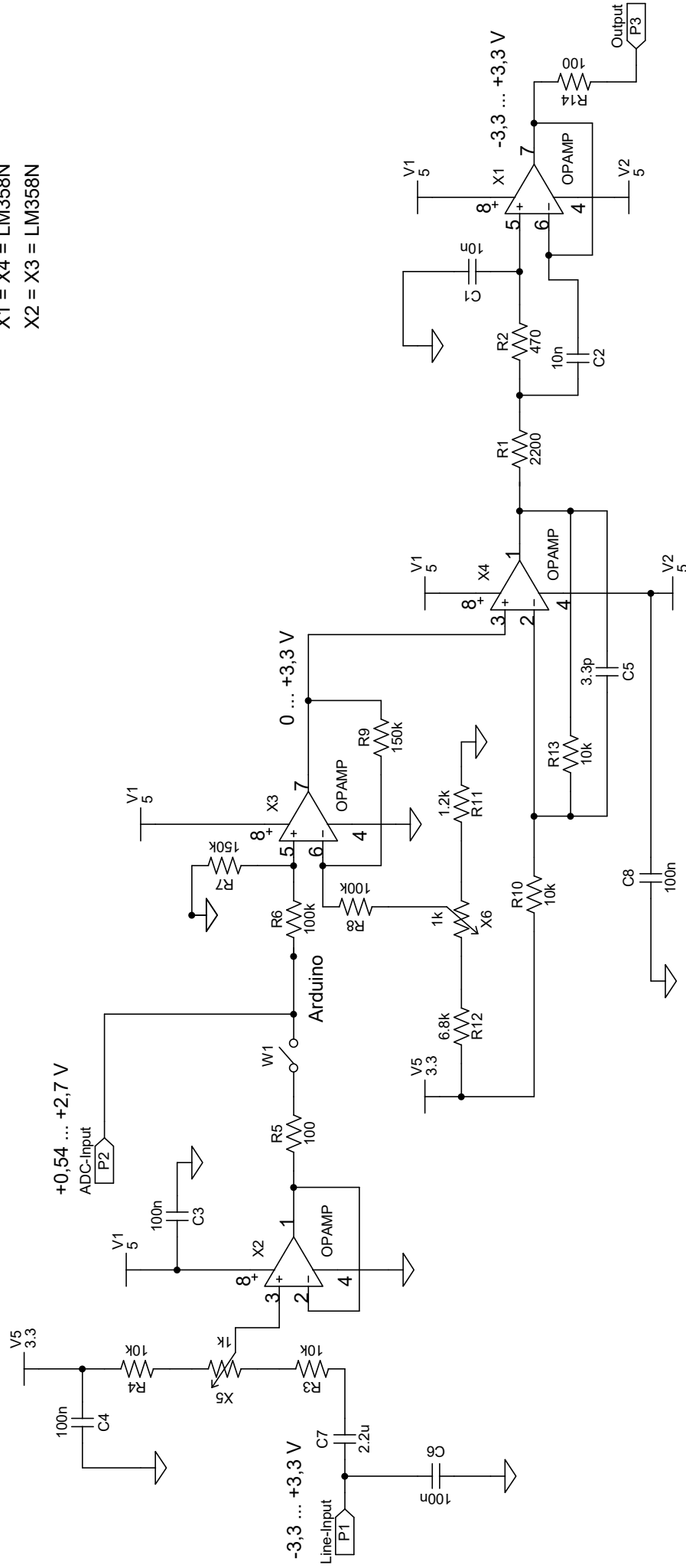
Page: 1 of 1    Size: A    HAW Hamburg

Rev: 90 06-Jul-2019 14:03    Created: 10-May-2019

Drawn by: Antonia Schwab

File: E:\Circuits\Senderschaltung.sch

X1 = X4 = LM358N  
 X2 = X3 = LM358N



Schaltung für den Verschlüsselungsempfänger

Page: 1 of 1 Size: A HAW-Hamburg

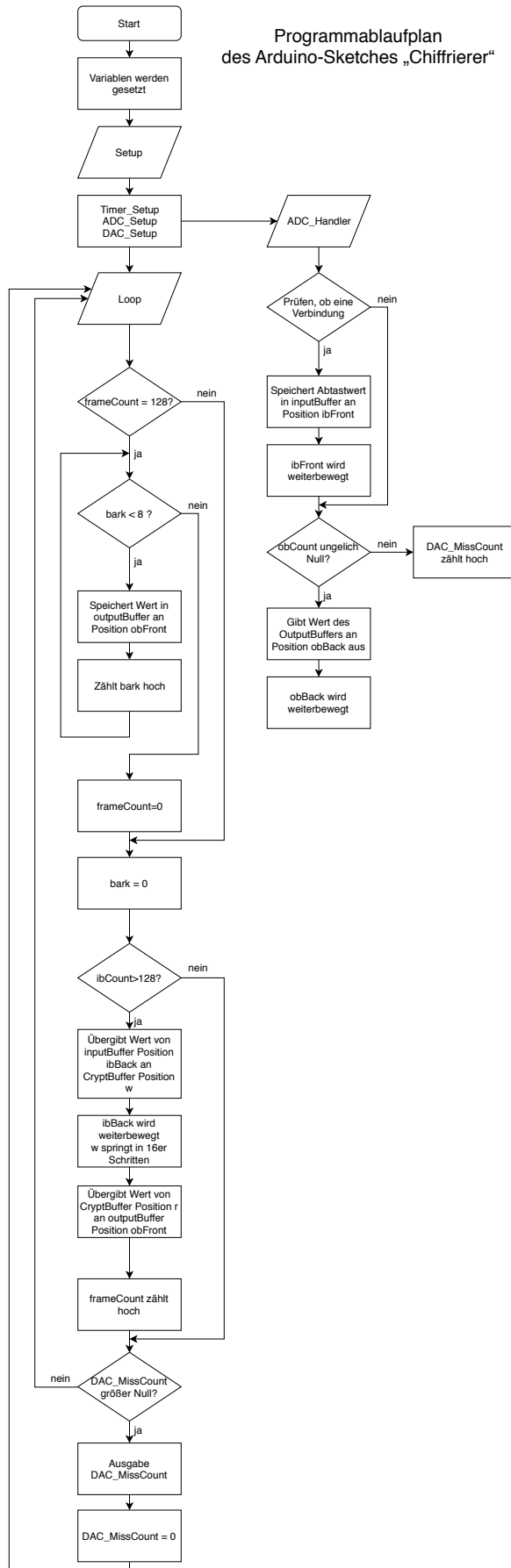
Rev: 81 03-Jul-2019 11:49 Created: 12-May-2019

Drawn by: Antonia Schwab

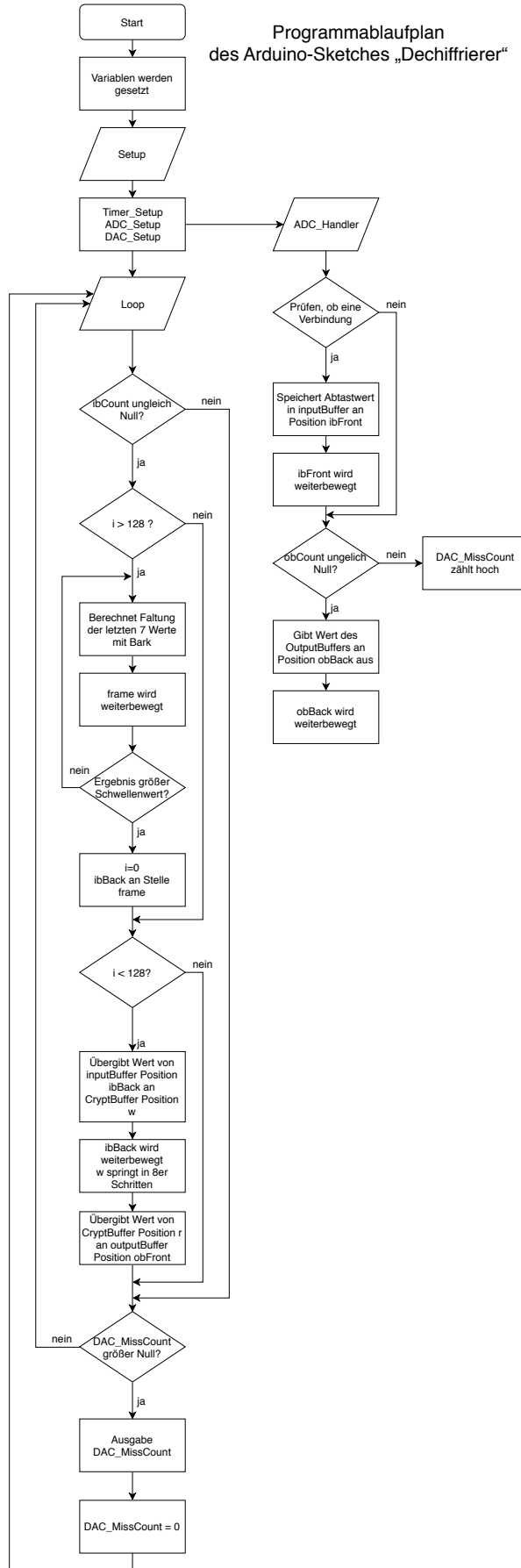
File: E:\Circuits\Empfängerschaltung.sch

## **A.4 Programmablaufpläne der Sketche Chiffrierer und Dechiffrierer**

Programmablaufplan  
des Arduino-Sketches „Chiffrierer“



Programmablaufplan  
des Arduino-Sketches „Dechiffrierer“





## **A.5 Fragebögen und Notizen zum Hörtest**

# Hörtest zur Validierung der Verschlüsselung Notizen

## 1. Testperson:

- A
- tieffrequente Signalanteile  $\rightarrow$  Tendenz zu m
  - Insgesamt Orientierung nur an Länge & Struktur
  - hat Struktur auch in F gehört
- $\rightarrow$  Inhalt ~~der~~ Sätze des Satzes A & B mitgeteilt
- $\hookrightarrow$  danach nicht/nur begrenzt der Meinung, es wiedererkant zu haben

## 2. Testperson:

- Vorschlag, Sync-Sequenz öfter zu senden (Lange Schlüssel) und darin Info zu verpacken, welcher Abschnitt des Signals es ist
- überrascht über schlechte Audiopnalität von C
- Vorschlag, das System in R3 zu stellen, um damit Sound-design zu machen

## 3. Testperson:

- nur wenig Zeit gehabt, daher alle Files tatsächlich nur 1x gehört
- $\rightarrow$  daher vermutlich die Aussage, alle seien identisch gewesen (sonst wahrscheinlich A und B ausgedrückt?)
- $\hookrightarrow$  schwer zu beurteilen

## 4. Testperson:

- wusste zu Beginn nicht, dass es sich um verschlüsselte Sprachsignale handelt
- $\rightarrow$  konnte dies sehr schnell an der Struktur erkennen
- hat nur bei D und E ähnliche Strukturen gehört
- $\hookrightarrow$  Vermutung, dass nur diese beiden das selbe Signal seien
- wenig Verwissen über Verschlüsselung etc

## 5. Testperson:

- bei A "weiß ich nicht" angekennzeichnet, vermutet aber männlich
- ~~hat~~ vermutet, dass A & B und D & E das gleiche Signal waren

# Hörtest zur Validierung der Verschlüsselung

Bachelor-Thesis Antonia Schwab

**A**

1. Konntest du etwas verstehen?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

*Vermutung!*

3. Was hast du verstanden?

**B**

1. Konntest du etwas verstehen?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?

**C**

1. Konntest du etwas verstehen?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden? *Willehmen zur Präsentation*

# Hörtest zur Validierung der Verschlüsselung

Bachelor-Thesis Antonia Schwab

**D**

1. Konntest du etwas verstehen?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?

**E**

1. Konntest du etwas verstehen?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?

**F**

1. Konntest du etwas verstehen?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?

Weitere Anmerkungen: *Willehmen zur Präsentation*

Hörtest zur Validierung der Verschlüsselung

Bachelor-Thesis Antonia Schwab

A

1. Konntest du etwas verstehen?  Ja  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

3. Was hast du verstanden?  
*es handelt sich um Sprache*

B

1. Konntest du etwas verstehen?  Ja  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

3. Was hast du verstanden?  
*sele A*

C

1. Konntest du etwas verstehen?  Ja  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

3. Was hast du verstanden?  
*Wolkennormen aus Pöschelmann*

D

1. Konntest du etwas verstehen?  Ja  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

3. Was hast du verstanden?  
*DT U klangen Identisch*

*Alle anderen waren unterschiedlich, da das Sprache mehrere nicht passte*

Hörtest zur Validierung der Verschlüsselung

Bachelor-Thesis Antonia Schwab

E

1. Konntest du etwas verstehen?  Ja  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

3. Was hast du verstanden?  
*es handelt sich um Sprache*

F

1. Konntest du etwas verstehen?  Ja  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

3. Was hast du verstanden?  
*DT U klangen Identisch*

G

1. Konntest du etwas verstehen?  Ja  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

3. Was hast du verstanden?  
*Alle anderen waren unterschiedlich, da das Sprache mehrere nicht passte*

H

1. Konntest du etwas verstehen?  Ja  Nein

2. Ist die Stimme...  Männlich  Weiblich  Weiß ich nicht

3. Was hast du verstanden?  Ja, ohne Probleme  Ja, mit Anstrengung  Nur einzelne Teile  Nein

3. Was hast du verstanden?  
*DT U klangen Identisch*

*Alle anderen waren unterschiedlich, da das Sprache mehrere nicht passte*

# Hörtest zur Validierung der Verschlüsselung

Bachelor-Thesis Antonia Schwab

**A**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

*W.A. well*  
*W.A. rausgehört*

3. Was hast du verstanden?

**B**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

*fehlweise*  
*rausgehört*

3. Was hast du verstanden?

**C**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

*Willkommen zur Präsentation.*

*Julia 3,1*  
*Julia 730*

# Hörtest zur Validierung der Verschlüsselung

Bachelor-Thesis Antonia Schwab

**D**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

*W.A. well*  
*W.A. rausgehört*

3. Was hast du verstanden?

**E**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

**F**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

Weitere Anmerkungen:

*A, B ähnlich*  
*D, E, F ähnlich*

# Hörtest zur Validierung der Verschlüsselung

Bachelor-Thesis Antonia Schwab

**A**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

\_\_\_\_\_

**B**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

\_\_\_\_\_

**C**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

\_\_\_\_\_

**D**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

Ullmann zur Piktografie

# Hörtest zur Validierung der Verschlüsselung

Bachelor-Thesis Antonia Schwab

**D**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

\_\_\_\_\_

**E**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

\_\_\_\_\_

**F**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

\_\_\_\_\_

**G**

1. Konntest du etwas verstehen?

Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...

Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

\_\_\_\_\_

Weitere Anmerkungen:

# Hörtest zur Validierung der Verschlüsselung

Bachelor-Thesis Antonia Schwab

**A**

1. Konntest du etwas verstehen?  
 Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...  
 Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

**B**

1. Konntest du etwas verstehen?  
 Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...  
 Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

**C**

1. Konntest du etwas verstehen?  
 Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...  
 Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

**D**

1. Konntest du etwas verstehen?  
 Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...  
 Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

Weitere Anmerkungen:  
Willkommen zur Präsentation.

# Hörtest zur Validierung der Verschlüsselung

Bachelor-Thesis Antonia Schwab

**D**

1. Konntest du etwas verstehen?  
 Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...  
 Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

**E**

1. Konntest du etwas verstehen?  
 Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...  
 Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

**F**

1. Konntest du etwas verstehen?  
 Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...  
 Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

**G**

1. Konntest du etwas verstehen?  
 Ja, ohne Probleme  
 Ja, mit Anstrengung  
 Nur einzelne Teile  
 Nein

2. Ist die Stimme...  
 Männlich  
 Weiblich  
 Weiß ich nicht

3. Was hast du verstanden?

Weitere Anmerkungen:  
Aufgrund des versteckten Stücks kann die Länge der Clips, wurde ich darauf schließen, dass immer dasselbe gespielt wurde.

## **A.6 Auskunft zu Mikrofonfunkstrecken der Firmen Shure und Beyerdynamic per E-Mail**



**Von:** support@shure.de  
**Betreff:** AW: Abschlussarbeit - Fragen zu Funkstrecken: [ ref:\_00D11Uyte.\_5001ILD1z4:ref ]  
**Datum:** 19. Juni 2019 um 13:23  
**An:** [REDACTED]



Hallo Antonia,

sehr interessantes Thema - wobei ich spontan sagen würde FM lässt sich nicht verschlüsseln. Aber da gibt es sicherlich auch sehr schlaue Köpfe, die sich da was überlegt haben.

Zu unseres verschlüsselbaren Funksystemen (QLXD, ULXD, Axient Digital). Diese sind alle Digital und arbeiten mit der 4PSK (Phase Shift Keying) bzw. mit 16QAM (Axient Digital)

Und aj - der 265 Bit Schlüssel wird über Infrarot übertragen.

Hoffe das hilft dir weiter. Für weiter Fragen stehe ich gerne zur Verfügung.

Beste Grüße

-----  
Jürgen Schwörer  
Senior Applications Engineer  
Shure Distribution GmbH  
Jakob Dieffenbacher Str. 12  
75031 Eppingen  
Germany

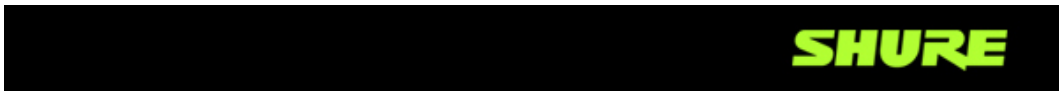
----- Ursprüngliche Nachricht -----

**Von:** info@shure.de [cspnotification@shure.com]

**Gesendet:** 19.06.2019 11:37

**An:** [REDACTED]

**Thema:** Abschlussarbeit - Fragen zu Funkstrecken:



shure.de



Ihre Frage ist eingegangen.

Sie können innerhalb von 24 Stunden mit einer Antwort von uns rechnen.

Fallbetreff: Abschlussarbeit - Fragen zu Funkstrecken

Datum erstellt: 6/19/2019

Datum der letzten Aktualisierung: 6/19/2019

Status: Assigned

Fallnummer: 00407214

Betreff: Sehr geehrtes Shure-Team,

ich bin Studentin der Medientechnik an der HAW Hamburg und schreibe im Moment meine Abschlussarbeit mit dem Titel "Ein Echtzeitverschlüsselungssystem für Mikrofonfunkstrecken". Ziel der Arbeit ist es, ein System zu entwickeln, das in Zusammenspiel mit Mikrofonfunkstrecken, die mittels FM übertragen, das Audiosignal vor dem Sender zu verschlüsseln und nach dem Empfänger wieder zu entschlüsseln.

Für meine Recherche habe ich jetzt einige Fragen an Sie, die besonders die Systeme betreffen, die bereits eine Verschlüsselung mit anbieten, und hoffe, dass Sie darüber Auskunft geben können.


1. Welche Modulationsverfahren werden bei digital arbeitenden Funksystemen verwendet? Sind diese unterschiedlich oder hat sich ein bestimmtes Verfahren als besonders vorteilhaft herausgestellt?

2. Wird der Schlüssel für die AES-Verschlüsselung, sofern sie verwendet wird, über die Infrarot-Schnittstelle bei der Synchronisierung mit übertragen oder gibt es einen anderen Weg?

Ich würde mich sehr über eine Antwort freuen und bedanke mich bei Ihnen schon im Voraus.

Sollten Sie noch Fragen haben, erreichen Sie mich auch telefonisch unter [REDACTED].

Mit freundlichen Grüßen,  
Antonia Schwab

**Von:** Gmoser, Tobias Tobias.Gmoser@beyerdynamic.de   
**Betreff:** WG: Antwort zu C00017739 Kontaktformular DE [ ref: \_00D1tDL3j\_5001tCF7lw:ref ]  
**Datum:** 25. Juni 2019 um 09:13  
**An:** [REDACTED]  
**Kopie:** Pietschmann, Michael Michael.Pietschmann@beyerdynamic.de



Hallo Frau Schwab,

Schön zu hören, dass Sie sich mit dem Themenfeld digitale Drahtlosmikrofone befassen. Um auf Ihre Fragen zu antworten:

1. Es kommen verschiedene Modulationsverfahren zum Einsatz. Es gibt sowohl Systeme, die FSK-Verfahren verwenden, als auch Systeme die auf Quadraturmodulation (QPSK oder QAM) basieren. In unserem TG1000-System zum Beispiel kommt ein FSK-Verfahren zum Einsatz, damit können wir eine sehr große Schaltbandbreite bei geringem Stromverbrauch in den Sendern realisieren.
2. Ich kann für unser System reden, hier wird der Schlüssel tatsächlich über die Infrarot-Schnittstelle ausgetauscht. Ich gehe aber davon aus, dass das bei Systemen anderer Hersteller ebenso gemacht wird. Infrarot ist nach wie vor die primäre Schnittstelle zur Konfiguration der Sender.

Ich hoffe, ich konnte Ihnen damit weiterhelfen. Gibt es eine Möglichkeit, nach Fertigstellung eine Abschrift Ihrer Arbeit zu erhalten?

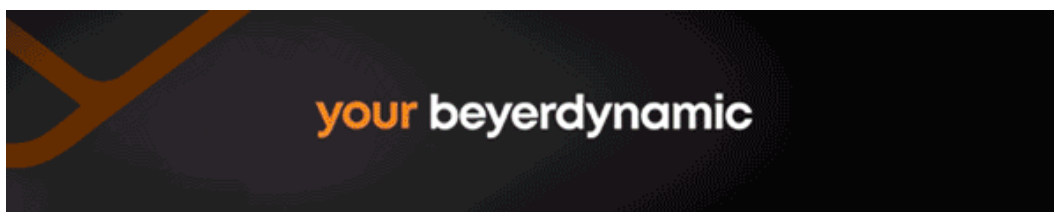
Mit freundlichen Grüßen,  
Tobias Gmoser

**TOBIAS Gmoser**  
TEAM MANAGER PRODUCT DEVELOPMENT  
DIVISION INSTALLED SYSTEMS / R&D

T + 49 7131 617-303  
F + 49 7131 617-215  
[tobias.gmoser@beyerdynamic.de](mailto:tobias.gmoser@beyerdynamic.de)  
[www.beyerdynamic.com](http://www.beyerdynamic.com)

**your beyerdynamic**

beyerdynamic GmbH & Co. KG · Theresienstraße 8 · 74072 Heilbronn - Germany  
Registergericht: Amtsgericht Stuttgart · HRA 101400 · Sitz der Gesellschaft: Heilbronn  
Geschäftsführer: Edgar van Velzen  
Allgemeine Geschäftsbedingungen: [hier klicken](#)



-----Ursprüngliche Nachricht-----

Von: [REDACTED]  
Telefon: [REDACTED]

Sehr geehrtes Beyerdynamic-Team,

ich bin Studentin der Medientechnik an der HAW Hamburg und schreibe im Moment meine Abschlussarbeit mit dem Titel "Ein Echtzeitverschlüsselungssystem für Mikrofonfunkstrecken". Ziel der Arbeit ist es, ein System zu entwickeln, das in Zusammenspiel mit Mikrofonfunkstrecken, die mittels FM übertragen, das Audiosignal vor dem Sender zu verschlüsseln und nach dem Empfänger wieder zu entschlüsseln.

Für meine Recherche habe ich jetzt einige Fragen an Sie, die besonders die Systeme betreffen, die bereits eine Verschlüsselung mit anbieten, und hoffe, dass Sie darüber Auskunft geben können.

1. Welche Modulationsverfahren werden bei digital arbeitenden Funksystemen verwendet? Sind diese unterschiedlich oder hat sich ein bestimmtes Verfahren als besonders vorteilhaft herausgestellt?

2. Wird der Schlüssel für die AES-Verschlüsselung, sofern sie verwendet wird, über die Infrarot-Schnittstelle bei der Synchronisierung mit übertragen oder gibt es einen anderen Weg?

Ich würde mich sehr über eine Antwort freuen und bedanke mich bei Ihnen schon im Voraus. Sollten Sie noch Fragen haben, erreichen Sie mich auch telefonisch unter [REDACTED].

Mit freundlichen Grüßen,  
Antonia Schwab

**MICHAEL PIETSCHMANN**  
KUNDENSERVICE / CUSTOMER CARE

T 00800 770 880 99  
T +49 7131 617-300  
[help@beyerdynamic.com](mailto:help@beyerdynamic.com)  
[www.beyerdynamic.com](http://www.beyerdynamic.com)

**your beyerdynamic**

beyerdynamic GmbH & Co. KG · Theresienstraße 8 · 74072 Heilbronn - Germany  
Registergericht: Amtsgericht Stuttgart · HRA 101400 · Sitz der Gesellschaft: Heilbronn  
Geschäftsführer: Edgar van Velzen  
Allgemeine Geschäftsbedingungen: [hier klicken](#)



# Abbildungsverzeichnis

1.1	Schematische Darstellung des geplanten Systemaufbaus . . . . .	6
2.1	Scramblingstruktur bei einer Blocklänge von 4 Werten . . . . .	10
2.2	Vergleich der Spektren des Originalsignals F3 mit der Verschlüsselung des Signals mit Blocklänge 4 . . . . .	10
2.3	Vergleich der Spektren des Originalsignals F3 mit dem der Verschlüsselung des Signals mit Blocklänge 128 . . . . .	11
2.4	Darstellung der Verschlüsselungsstruktur bei einer Blocklänge von 128 Werten . . . . .	12
2.5	Vergleich der Spektren des Originalsignals F3 mit der Verschlüsselung des Signals mit Blocklänge 128 und Verschachtelung innerhalb des Blockes . . . . .	12
2.6	Vergleich der Spektren des mit unterschiedlichen Blocklängen verschlüsselten Signals F3-Original . . . . .	14
2.7	Vergleich des mit unterschiedlichen Blocklängen verschlüsselten Signals F3 im Zeitbereich. Links: Original, Mitte: Schlüssel 128, Rechts: Schlüssel 1024 . . . . .	15
2.8	Autokorrelation der Barker-Codesequenz der Länge 7 . . . . .	17
2.9	Systemstruktur der Senderseite . . . . .	17
2.10	Systemstruktur der Empfängerseite . . . . .	18
2.11	Vergleich der Spektren des Originalsignals F3 mit dem verschlüsselten und entschlüsselten Signal (OTP) nach Kanalübertragung . . . . .	20
2.12	Vergleich des Originalsignals F3 mit dem entschlüsselten Signal (OTP) nach Kanalübertragung im Zeitbereich . . . . .	21
2.13	Links: Spektrum des verschlüsselten Signals (128), Rechts: Spektrum des Signals nach Senden über Kanal . . . . .	22
2.14	Periodische Barker-Codesequenz im Zeit- und Frequenzbereich . . . . .	23
2.15	Spektrum des verschlüsselten Signals mit Schlüssel 128 und der Barker-Codesequenz . . . . .	23
2.16	Darstellung der Verschlüsselungsstruktur bei einer Blocklänge von 128 Werten . . . . .	24
2.17	Spektrum des verschlüsselten Signals mit Schlüssel 128 und der Barker-Codesequenz . . . . .	24
3.1	Foto des Chiffrierers . . . . .	31
3.2	Schematischer Aufbau des Programms . . . . .	32

## Abbildungsverzeichnis

3.3	Timer auf Sender- und Empfängerseite mit einer Frequenz von 30 kHz	33
3.4	Programmierung der Synchronisierung des Chiffrierers . . . . .	34
3.5	Programmierung der Synchronisierung des Dechiffrierers . . . . .	35
3.6	Screenshot des seriellen Plotters, Ausgabe der Variable y zur Einstellung des Schwellenwertes . . . . .	37
3.7	Buffer-Struktur auf Senderseite am Beispiel des Schlüssels 128 . . . . .	37
3.8	links: exemplarische Chiffrierung mit skaliertem Barker-Codesequenz, rechts: exemplarische Dechiffrierung . . . . .	39
3.9	Voltage Follower Pulse Response des LM358-N ( <a href="#">Texas Instruments 2014</a> )	39
3.10	Barker-Codesequenz am Output der Schaltung (Kanal 1) und direkt am DAC (Kanal 2) . . . . .	40
3.11	Signalstörung nach Auseinanderlaufen der Timer, Kanal 1: DAC des Chiffrierers, Kanal 2: DAC des Dechiffrierers . . . . .	41
3.12	Messung der Eingangswerte des ADC bei Anlegen einer konstanten Spannung in Höhe von 0,6 V . . . . .	43
4.1	Messaufbau für Messungen mit dem D-Scope 3 der Firma Prismasound	45
4.2	links: Amplitudenfrequenzgang des Chiffrierers inklusive Mikrofonvorverstärker, rechts: Amplitudenfrequenzgang des Dechiffrierers . . . . .	46
4.3	Amplitudenfrequenzgänge mit unterschiedlichen Schlüsseln . . . . .	46
4.4	Shure PGX: Phasenfrequenzgang . . . . .	47
4.5	Phasenfrequenzgänge des Chiffrierers (oben), des Dechiffrierers (mitte) und der gesamten Verschlüsselungseinheit (unten) . . . . .	49
4.6	Gesamtsystem: Phasenfrequenzgang . . . . .	50
4.7	Verschlüsselung 128: Phasenfrequenzgang 500 bis 1000 Hz . . . . .	51
4.8	Verschlüsselung 256: Phasenfrequenzgang 400 bis 500 Hz . . . . .	52
4.9	Verschlüsselung 512: Phasenfrequenzgang 400 bis 500 Hz . . . . .	52
4.10	Verschlüsselung 1024: Phasenfrequenzgang 400 bis 500 Hz . . . . .	53
4.11	Messung der Latenz mit Oszilloskop für Schlüssel 128 . . . . .	54
4.12	Messung der Latenz mit Oszilloskop für Schlüssel 256 . . . . .	55
4.13	Messung der Latenz mit Oszilloskop für Schlüssel 512 . . . . .	56
4.14	Messung der Latenz mit Oszilloskop für Schlüssel 1024 . . . . .	57
4.15	Shure PGX: THD+N . . . . .	59
4.16	Chiffrierer: THD+N . . . . .	60
4.17	Dechiffrierer: THD+N . . . . .	60
4.18	Verschlüsselung 128: THD+N . . . . .	61
4.19	Verschlüsselung 256: THD+N . . . . .	62
4.20	Verschlüsselung 512: THD+N . . . . .	62
4.21	Verschlüsselung 1024: THD+N . . . . .	63
4.22	Eingangswerte am Verschlüsselungsempfänger bei Nutzung der Funkübertragungsstrecke . . . . .	65
4.23	Spannung am Ausgang des Funkempfängers PGX4 von Shure . . . . .	65

*Abbildungsverzeichnis*

4.24	Autokorrelation der gemessenen Werte der Barker-Codesequenz nach der Übertragung über die Mikrofonfunkstrecke . . . . .	66
4.25	Übersicht der Ergebnisse des Hörtests für Frage 1 . . . . .	70
4.26	Übersicht der Ergebnisse des Hörtests für Frage 2 . . . . .	71

# Tabellenverzeichnis

2.1	Latenzen bei unterschiedlichen Scrambling-Blocklängen und der Abtastrate $f_s = 30000$ Hz . . . . .	16
2.2	Vergleich unterschiedlicher Abtastraten für verschiedene Blocklängen des Chiffrierers . . . . .	19
4.1	Gemessene Latenzen bei Verwendung unterschiedlicher Schlüssel (D-Scope: berechnet aus Phasenfrequenzgang, M1 bis M3; gemessen mit Oszilloskop) . . . . .	58

# Literaturverzeichnis

- Atmel: *SAM3X / SAM3A Series. Atmel | SMART ARM-based MCU. Datasheet*, [http://ww1.microchip.com/downloads/en/devicedoc/atmel-11057-32-bit-cortex-m3-microcontroller-sam3x-sam3a\\_datasheet.pdf](http://ww1.microchip.com/downloads/en/devicedoc/atmel-11057-32-bit-cortex-m3-microcontroller-sam3x-sam3a_datasheet.pdf), 2015, letzter Zugriff: 3. 7. 2019
- Baunetz Wissen: *Mono-Mikrofonvorverstärker*, <https://www.baunetzwissen.de/akustik/fachwissen/schallreflexion/echo-147761>, 2015, letzter Zugriff: 5. 7. 2019
- Beutelspacher, Albrecht: *Kryptologie. Eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen*, 10. Auflg., Springer-Verlag 2015
- Carrera, Giovanni: *A bipolar analog I/O for Arduino Due*, <http://ardupiclab.blogspot.com/2015/10/a-bipolar-analog-io-for-arduino-due.html>, 2015, letzter Zugriff: 4. 7. 2019
- Baunetz Wissen: *Echo*, [https://produktinfo.conrad.com/datenblaetter/175000-199999/197688-as-03-de-Mono\\_Mikrofon\\_Vorverstaerker.pdf](https://produktinfo.conrad.com/datenblaetter/175000-199999/197688-as-03-de-Mono_Mikrofon_Vorverstaerker.pdf), 1999, letzter Zugriff: 6. 7. 2019
- Dembowski, Klaus: *Mikrocontroller - Der Leitfaden für Maker. Schaltungstechnik für Raspberry, Arduino & Co.*, 1. Auflg., DPunkt 2014
- Dennis, Ian: *D-Scope Series 3. Operation Manual*, [http://resources.prismsound.com/tm/dS3\\_Operation\\_Manual\\_A4nc.pdf](http://resources.prismsound.com/tm/dS3_Operation_Manual_A4nc.pdf), 2012, letzter Zugriff: 4. 7. 2019
- DIN EN 61606-3: *Audio- und audiovisuelle Geräte - Digitale Tonteile - Grundlegende Messverfahren der Audio-Eigenschaften - Teil 3: Professioneller Gebrauch (IEC 61606-3:2008); Deutsche Fassung EN 61606-3:2008*, <https://secure-1beuth-1de-10000017m3896.shan02.han.tib.eu/cmd%3Bjsessionid=1112HGRKNF4K1AP5E3IUS1XD.4?workflowname=instantdownload&customerid=328165&docname=1505403&contextid=eeas&servicerefname=eeas&LoginName=bvollbrecht1>, 2009, letzter Zugriff: 4. 7. 2019
- Ertel, Wolfgang & Löhmann, Ekkehard: *Angewandte Kryptographie*, 5. Auflg., Hanser 2018
- Gessler, Ralf: *Entwicklung eingebetteter Systeme*, Springer-Verlag 2014



## Literaturverzeichnis

- Gmoser, Tobias: *Mailauskunft. WG: Antwort zu C00017739 Kontaktformular*. [Tobias.Gmoser@beyerdynamic.de], 25.06.2019
- Görne, Thomas: *Tontechnik*, 3. Aufl., Hanser 2011
- Goertz, Anselm & Schmitz, Alfred: *Raumakustik und Sprachverständlichkeit*, <http://www.ifaa-akustik.de/files/zvei-ens-saa-2012-04-26-a-goertz.pdf>, 2012, letzter Zugriff: 6. 7. 2019
- Meyer, Martin: *Kommunikationstechnik. Konzepte der modernen Nachrichtenübertragung*, 5. Aufl., Springer-Verlag 2014
- Müller, Swen: „Messtechnik“, in: Weinzierl, Stefan (Hrsg.): *Handbuch der Audiotechnik*, Springer-Verlag 2008
- Newton, Chris: *ADC to DAC on the Arduino*, <https://cjpnmiscellany.wordpress.com/2015/01/28/adc-to-dac-on-the-arduino/>, 2015, letzter Zugriff: 3. 7. 2019
- Niehoff, Wolfgang: „Drahtlose Übertragungstechnik“, in: Weinzierl, Stefan (Hrsg.): *Handbuch der Audiotechnik. Band 2*, Springer-Verlag 2008
- Paar, Christof & Pelzl, Jan: *Kryptographie verständlich erklärt*, Springer-Verlag 2016
- Schneider, Martin: „Mikrofone“, in: Weinzierl, Stefan (Hrsg.): *Handbuch der Audiotechnik*, Springer-Verlag 2008
- Schwörer, Jürgen: *Mailauskunft. AW: Abschlussarbeit - Fragen zu Funkstrecken*. [support@shure.de], 19.06.2019
- Sertronics: *Arduino Due, Datenblatt*, <https://www.berrybase.de/Pixelpdfdata/Articlepdf/id/1590/onumber/A000062>, 2019, letzter Zugriff: 3. 7. 2019
- Shure: *PGX Funksysteme*, <https://www.shure.de/productdocumentsfiles/default/discontinued/wireless/pgxdatenblatt-deutsch--86f151e74fce390580f5f223bbcbf663.pdf>, 2010, letzter Zugriff: 8. 7. 2019
- Shure: *SM WIRED MICROPHONES*, <https://pubs-api.shure.com/file/260007>, 2014, letzter Zugriff: 4. 7. 2019
- ST Microelectronics: *TSH110-111-112-113-114. Wide Band, Low Noise Operational Amplifiers*, <https://www.digchip.com/datasheets/parts/datasheet/456/TSH112-pdf.php>, 2002, letzter Zugriff: 7. 7. 2019
- ST Microelectronics: *Barker Code*, <http://mathworld.wolfram.com/BarkerCode.html>, 2019, letzter Zugriff: 7. 7. 2019

## Literaturverzeichnis

Texas Instruments: *LMx58-N Low-Power, Dual-Operational Amplifiers*, <http://www.ti.com/lit/ds/symlink/lm358-n.pdf>, 2014, letzter Zugriff: 7. 7. 2019

Umweltbundesamt: *Schalldruckpegel, energieäquivalenter Dauerschallpegel und Lärmindizes*, <https://www.umweltbundesamt.at/umweltschutz/laerm/schalldruckpegel>, 2014, letzter Zugriff: 4. 7. 2019

Wätjen, Dietmar: *Kryptographie. Grundlagen, Algorithmen, Protokolle*, 3. Aufl., Springer-Verlag 2018

Ich versichere, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangaben eindeutig kenntlich gemacht.

Ort, Datum

Antonia Schwab