UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science
Information Technology

Maksim Lind

Secure Data Transmission over Mobile Voice Channel

Bachelor's Thesis (6 ECTS)

Supervisor: Alexander Tkachenko, M.Sc.

TARTU 2015

Secure Data Transmission over Mobile Voice Channel

Abstract:

Nowadays mobile communication is one of the most common forms of exchanging information. Although mobile communication standards were designed with security in mind, a few known vulnerabilities exist. An attacker can penetrate a mobile network and eavesdrop on a victim's conversation or access data servers of the operator where conversation recordings are stored.

A number of attempts has been done to address the issue of mobile communication security. Software solutions use a mobile data channel with standard security mechanisms for information exchange. Hardware solutions represent custom phones that create secure voice channel between two such devices. The first solution does not suite users, who do not have access to the mobile data channel. The disadvantage of the second solution is a high cost and an incompatibility with general phones.

In this work, we describe an alternative solution, where security is enforced before any information reaches the phone. Sensitive information such as voice is processed in an external device and then passed into the mobile phone as an analog sound signal. The advantage of this approach is that the external unit and can be attached to any phone with a sound input.

While building the system, we analyzed a number of existing solutions, tuned parameters and performed experiments. As a result, we came up with a system that performs secure data transfer at a speed of up to 2000 bps and a median error rate of 21 percent.

Keywords:

Data transmission, voice channel, mobile communication, modulation, encryption.

Turvaline andmeedastus üle mobiilside häälekanali

Lühikokkuvõte:

Tänapäeval on mobiilside üks levinumaid viise informatsioonivahetuseks peale silmitsi suhtlemise ja interneti. Kuigi mobiilside standardid tagavad kõnede turvalisuse, on neis ka teadaolevaid turvaauke. Kindlates olukordades võib kurjategija mobiilikõnet pealt kuulata või pääseda juurde mobiilioperaatori poolt salvestatud kõnedele.

Mõned turvalise mobiilikõne lahendused kasutavad andmeedastuseks internetti. Teised pakkuvad spetsiaalseid telefone, mis loovad omavahel turvalise häälekanali. Esimene variant ei sobi inimestele, kellel pole ligipääsu mobiilsele internetile, teine variant on aga kallis ning seda ei saa kasutada teiste telefonidega.

Selles töös me kirjeldame lahendust, mis tagab turvalisuse enne informatsiooni telefonisse jõudmist. Tundlikud andmed töödeldakse eraldiseisvas seadmes ning saadetakse telefonisse analoogse helisignaalina. Sellist seadet saab ühendada iga telefoniga, millel on ette nähtud helisisend.

Töö käigus me vaatlesime erinevaid olemasolevaid lahendusi, seadistasime süsteemi parameetreid ja viisime läbi katsed. Tulemusena me saavutasime süsteemi, mis edastab turvaliselt andmed kiirusega kuni 2000 biti sekundis ja veamäära mediaaniga 21 protsenti.

Võtmesõnad:

Andmeedastus, häälekanal, mobiilside, modulatsioon, krüpteerimine.

Contents

Introduction

The secrecy of correspondence is a fundamental legal principle granted by the constitutions of most developed countries. It guarantees that sealed letters in transit will not be opened by authorities or any other third party. Nowadays, the secrecy of correspondence also extends to other media of communication, such as mobile network and Internet [kon12].

In mobile communication, the secrecy of correspondence is enforced by security standards that prevent third party access to private conversations or data. However, as these standards become obsolete, they become vulnerable to attackers [O'B09]. This creates a risk of a man-in-the-middle attack [FJHT14], [SB13].

Another security threat is related to an obligation of mobile operators to store communication data of their clients' data. If not securely stored, the data can be accessed by an unauthorized person.

One way to provide mobile conversation security is to make sure that information is secure along its way from one client to the other. This can be achieved by encrypting conversation data on a client side before it enters a mobile network. It prevents man-in-the middle attacks and unauthorized acces to data stored by a service provider.

A number of attempts has been made to address the issue of mobile communication security. Software solutions, such as smart phone applications, cannot acces baseband firmware to send data over mobile voice channel. Instead, they use a mobile data channel to access Internet and transfer secure data. This approach does not suit users who do not have access to the mobile data channel. Other solutions represent phones with the custom firmware that use mobile voice channel, but cannot connect to a usual phone.

In this work, we describe an alternative solution, where security is enforced before any information reaches the phone. Sensitive information such as voice is processed in an external device and then passed into the mobile phone as an analog sound signal. The advantage of this approach is that the external unit and can be attached to any phone with the sound input connector. In this thesis we set out to assemble a system capable of securely transmitting data over a mobile voice channel.

This thesis is organized as follows: Chapter 1 provides background informa-

tion on mobile voice channel, modulation, digital sound recording and encryption. Chapter 2 gives an overview of existing mobile security products and modulation software. Related work on data transmission over mobile voice channel is discussed. Chapter 3 describes a system prototype and an experimentation setup. Results and future work are described in Chapter 4.

Chapter 1

Background

1.1 Mobile Voice Channel

In 1987 a standard describing protocols for second generation (2G) digital cellular networks was developed. The standard was called GSM and was meant to replace the first generation analog cellular networks (1G). According to the first generation standard, an analog voice signal was modulated into a higher frequency and sent over the network. In contrast to that, the GSM specified that voice is to be processed into digital data and only then modulated into a carrier signal and sent over the air. Additionally, the GSM standard also specifies that digital data can be encrypted, but does not enforce it [RWO98]. The process of converting voice signal into digital data is described in section 1.3.

Every mobile phone has a module responsible for creating digital data stream from input speech and sending it to a mobile access point during a phone call. It can only be accessed by a vendor firmware of the phone and not by any software in case of a smart phone. In these circumstances, the mobile phone can be described as a black box that receives audio signal on one end and sends a signal that carries digital data out of the other end. Other standards were developed to support data transfer from mobile phones, introducing a mobile data channel.

As for the voice channel of the GSM and its successors, it is meant to transfer only speech. In order to increase a channel capacity, certain assumptions are made about the originating audio signal [RWO98]. A typical constraint is that the input signal is processed only at frequencies between 300 and 3400 Hz. This is a frequency range of a human voice.

Furthermore, a mobile voice channel and especially a modern one, has even more constraints for the input voice signal. Firstly, modern mobile channels and smart phones have a capability to detect noise and cancel it out on a software level. In some smart phones, this feature can be disabled. Secondly, the GSM introduces a voice activity detection system, which detects if the input signal contains human

speech. If not, the transmit cycles on sending side are reduced in order to save battery life time. The receiver side then generates a comfort noise [RWO98].

These constraints provide a better phone call experience, but make it hard to use mobile voice channel outside of its usual use case.

1.2 Modulation

In signal processing, modulation refers to a process of modifying a carrier signal so that it delivers some information. In digital modulation, a stream of bits is transferred over an analog channel. The carrier signal is usually a periodic waveform. A modulator is a device that performs modulation. A demodulator is a device that performs demodulation. A device that can both MOdulate and DEModulate is called a MODEM.

1.2.1 Basic Modulation Methods

There are several types of digital modulation. The most simple is On-Off keying (OOK). One bit of information is defined by presence or absence of a carrier signal.

Two widely used modulation methods are Amplitude-Shift Keying (ASK) and Frequency-Shift Keying (FSK). Their corresponding analog variants Amplitude Modulation (AM) and Frequency Modulation (FM) are commonly used in radio. In order to encode a single bit of data into a single frame of the carrier signal, ASK changes the amplitude of a carrier signal in this frame. Similarly, FSK changes the frequency of the carrier signal (refer to Figure 1.1).

A third basic type of modulation is Phase-Shift Keying (PSK). Here a phase shift of the signal is reversed when the state of binary data is changed (refer to Figure 1.2). Quadrature Amplitude Modulation (QAM) is a combination of PSK and ASK. In one frame, both amplitude and phase of a carrier signal may be changed.

1.2.2 Symbols and Constellations

A convenient way to describe a state of a signal is using a constellation diagram. A certain state of the signal is called a symbol and is described by the phase and the amplitude of the signal at a given moment. In a constellation diagram, the phase is represented by an angle around a circle and the amplitude by the distance from the center of the circle.

For Binary PSK (BPSK) two points appear on a constellation diagram: one for symbol "1" with phase shifted 180 degrees and another for symbol "0" with phase shifted 0 degrees (refer to Figure 1.3). Constellations of binary FSK and binary ASK also contain two symbols.

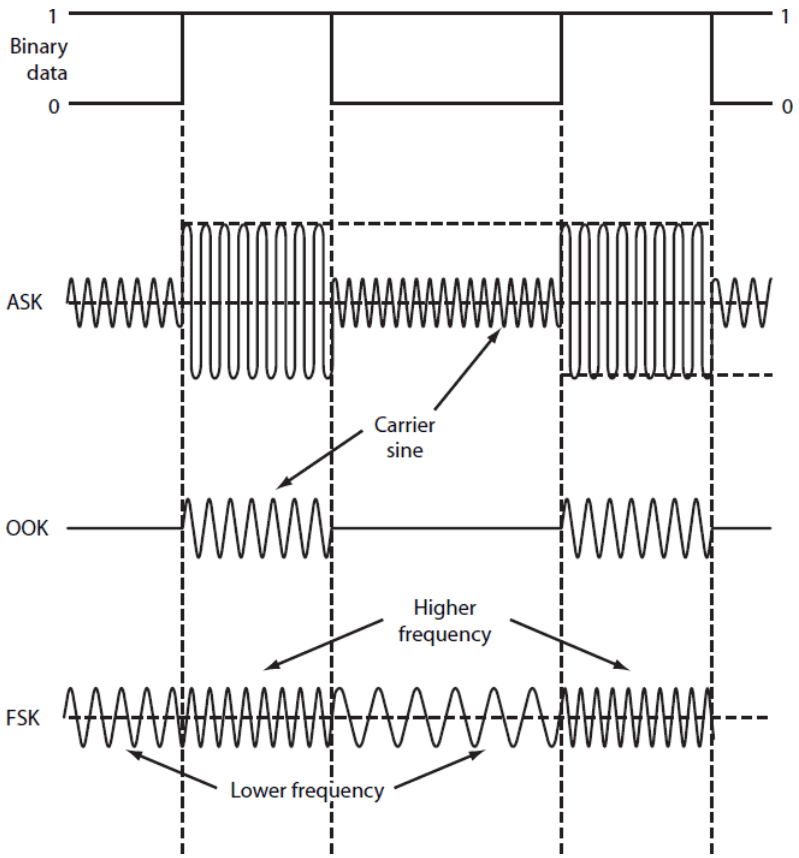

Figure 1.1: An illustration of ASK, OOK and FSK. To encode a bit "1" into a carrying signal, ASK increases an amplitude of the signal. For bit "0", an amplitude is decreased. FSK behaves similarly: for a bit "1" frequency is increased, for a bit "0" it is decreased. OOK defines the "1" bit as a presence of the signal and "0" as absence of it [Des12].

1.2.3 Complex Modulation Methods

By increasing a number of symbols in a constellation, it is possible to carry higher data rates. Quadrature PSK (QPSK) has four points in its constellation and each symbol contains two bits. It is achieved by making a phase-shift step 90 degrees instead of 180.

16-QAM has 16 points in a constellation, where each symbol consists of four bits (refer to Figure 1.4). Higher order versions of QAM are 32-QAM, 64-QAM, 128-QAM and 256-QAM.

Another way to achieve a higher bandwidth is multiplexing. Multiplexing is a method to combine multiple signals into one. Frequency-division multiplexing (FDM) achieves it by sending several signals in several distinct frequency ranges over a single medium (refer to Figure 1.5). This way, the constellation still has the same number of points, but multiple symbols are sent in one frame of time.

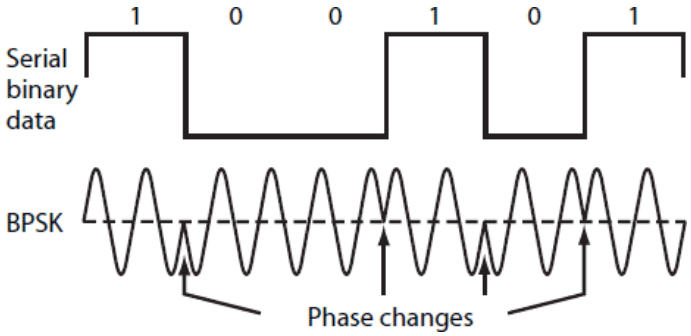

Figure 1.2: Phase-shift keying. PSK uses a phase of the carrier signal to encode data bits. To encode a change from 0 to 1 or from 1 to 0, the phase of signal is reversed in binary BSK [Des12].

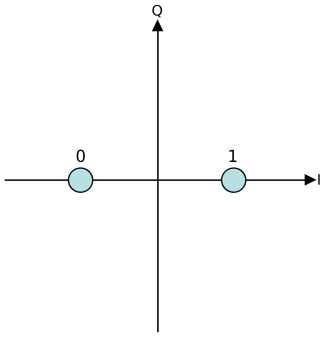

Figure 1.3: A binary phase-shift keying symbol constellation. In this constellation the distance from the center of the circle remains the same - amplitude of the carrier signal is unchanged. The phase of the signal is shifted by 180 degrees when a change in the binary signal occurs [uS06].

1.2.4 Comparison of Different Modulation Methods

Introducing more points in a constellation increases the number of bits per symbol. It also requires a demodulator to be more precise and immune to a channel noise. The more points there are in a constellation, the closer they are to each other.

Typically, usage of QAM is advised when there is a need for data rates above those that can be achieved using 8-PSK. The advantage of 8-PSK over QAM is that its points in a constellation have greater distance between each other. Another disadvantage of QAM is that the QAM demodulator has to detect both phase shift and amplitude of the signal. Table 1.1 illustrates different types of modulation.

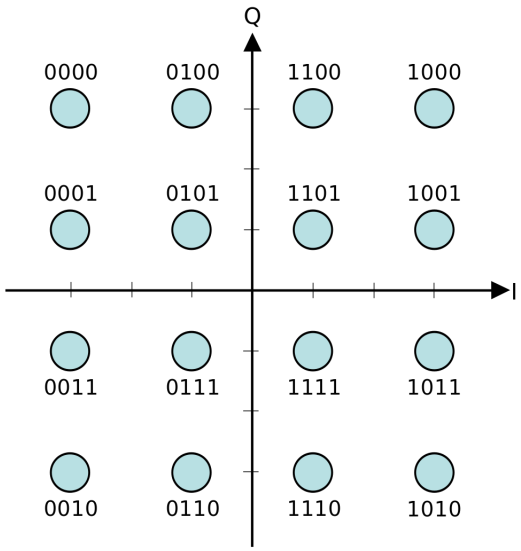

Figure 1.4: A symbol constellation of 16 Point Quadrature Amplitude Modulation (16-QAM). In this example four bits of data are modulated into a signal by simultaneously changing the carrier signal amplitude and shifting the signal phase [uS06].

Table 1.1: Comparison of different modulation methods. While higher order QAM increases data rate, it is also more susceptible to the noise [RE].

Modulation	Bits per symbol	Error margin	Complexity
OOK	1	$1/2 = 0.5$	Low
BPSK	1	$1 = 1$	Medium
QPSK	2	$1/\sqrt{2} = 0.71$	Medium
16-QAM	4	$\sqrt{2}/6 = 0.23$	High
64-QAM	6	$\sqrt{2}/14 = 0.1$	High

1.3 Digital Sound Recording

In digital sound recording, an analog sound signal is converted into a stream of digital data and these data are stored on a medium. An analog sound must be converted into discrete numbers that describe it.

1.3.1 Pulse-Code Modulation

Pulse-code modulation is a method of digitally representing sampled analog signals. In case of sound recording, PCM takes place right after a microphone has converted sound into electrical signal. An audio card receives this signal and needs to represent it as a stream of bits. To achieve that, an audio card "describes" the input audio signal at a set sampling rate with a set number of bits (refer to Figure 1.6). An audio codec specifies how exactly the audio signal is described.

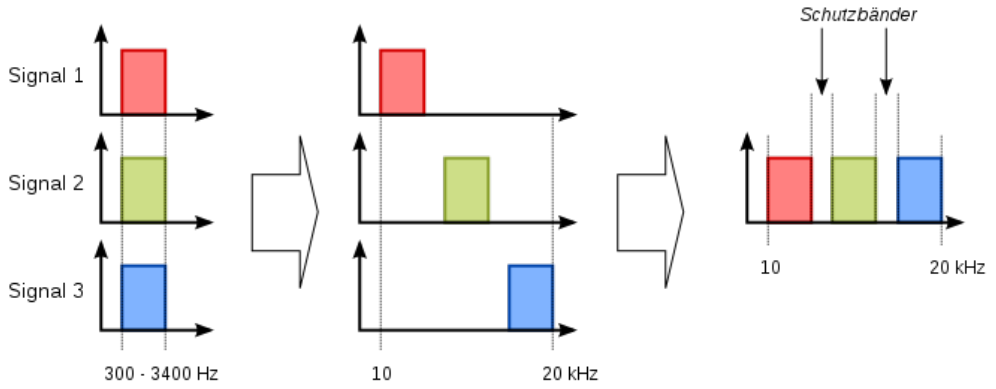

Figure 1.5: Frequency Division Multiplexing. In this example, three data-carrying signals are in frequency range 300-3400 Hz. These signals are modulated onto distinct frequencies in range 10-20 kHz and then sent simultaneously over one carrier medium [Boc12].

A standard audio sampling rate of a professional audio recording equipment is 44100 Hz. A common size of one sample is 16 bits. This means that the state of an input audio signal is described 44100 times per second with 16 bits of data. The higher the sample rate, the more accurate a digital representation of the audio signal is and the more storage is needed to save it. Sample rate of 8000 Hz is enough to adequately describe a human speech. On personal computers, digital audio data recorded using PCM is usually stored in WAV or RAW containers.

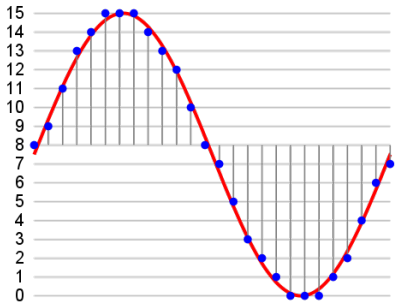

Figure 1.6: A 4-bit pulse code modulation. Analog signal is sampled and described detecting 16 different states [com14].

1.3.2 Audio Codec

Data compression allows using less storage for audio recordings. A program that compresses and decompresses digital audio data is called a codec. Audio compression can be divided into a lossless and lossy compression.

A lossless compression allows almost perfect reconstruction of original audio

signal. Popular lossless audio codecs are Free Lossless Audio Codec (FLAC) and Windows Media Lossless (WMA Lossless).

A lossy compression uses inexact approximation or partial data discarding to reduce amount of data that is needed to represent an audio signal. One of the most common examples of lossy compression usage in everyday life are MP3 files that store music on portable audio players. These files will not be suitable for professional audio engineering, but represent original sound well enough for a human's ear to not distinguish any inaccuracies.

1.3.3 Speech Codec

A speech codec is a lossy audio codec that is implemented to represent only human speech. Speech codec only processes sounds that could be made by a human voice and only keeps bare minimum of data from the original signal. Speech codecs are used in mobile communication. This is the reason why music over mobile conversation does not sound as good as when played back from an MP3 file.

Common speech codecs in mobile communication are Full Rate (GSM-FR) and Enhanced Full Rate (GSM-EFR) voice codecs. They extract the parameters characterizing a voice signal and represent them in a compact bit stream. At the receiver side, the replica of the original signal is constructed using these parameters [Mes03].

Speex [xosc] and codec2 [Row11] are open-source speech codecs. Codec2 is specialized in encoding data at a rate of 1200-3200 bps. Speex has a wider bit rate settings range available: 2000 to 44000 bps. Codec2 compresses a 96 KB raw audio file into four KB (less than five percent of original file).

1.4 Encryption

In cryptography, encryption is the process of encoding messages in a way, that prevents unauthorized parties to read it. Encryption does not prevent the message from getting into hands of a third party, but makes the message unreadable for the interceptor. Essential features of any encryption utility are a key-scheme and an encryption cipher.

1.4.1 Symmetric Key Encryption

In symmetric-key schemes, both communication parties must preliminary exchange an encryption key. The same key is used for encryption and decryption. If the encrypted message gets corrupted, the decryption key will mismatch and the original message cannot be restored.

1.4.2 Block Cipher

A block cipher is an algorithm, which operates on groups of bits called blocks. The blocks are encrypted separately using a symmetric or an asymmetric key. A block cipher can be used when sending encrypted data over a noisy channel and a data corruption is expected. If the data corruption occurs, it will only prevent restoring original message in corrupted blocks. One of the simplest and well-known block ciphers is a Caesar cipher, which encrypts text one character at a time.

1.4.3 Encryption utilities

Numerous free and open-source encryption and decryption utilities exist. The OpenSSL Project [ssl99] is a well-known full-strength general purpose cryptography library. It is robust and has a lot of features. Ccrypt [Sel00] utility focuses on encrypting and decrypting files and streams. Furthermore, it supports block cipher and can therefore decrypt corrupted files.

Chapter 2

Related Work

2.1 Existing Solutions for Secure Mobile Communication

Mobile voice channel security is a well-studied topic. Some off-the-shelf solutions are available that claim to achieve secure communication over a mobile channel. Software solutions like [CM] and [Sys14] encrypt voice calls, but cannot use a voice channel due to reasons mentioned in Section 1.1. Instead, they use Internet connection.

Hardware solutions like [Tri], [Com] and [Cry] represent custom mobile phones that can communicate securely with each other over mobile voice channel. Such phones are expensive and not compatible with normal mobile phones.

A reasonable compromise between software and hardware solutions is an external unit that processes the data before it reaches the phone and can be plugged into any phone like a headset. This allows utilizing mobile voice channel and increases the portability. Research in this direction is described in the following section.

2.2 Related Literature

Secure data transmission over a mobile voice channel has been studied from different aspects. [LSF08] describes a modem, which encodes digital data into a set of waveforms generated by a genetic algorithm. The modem then concatenates the pre-generated waveforms and sends them over the air. A genetic algorithm ensures that the waveforms are minimally distorted by a GSM-EFR codec. Such modem yielded raw data rate of four kbps and a symbol error rate of three percent. The disadvantage of this approach is that it is codec-specific. To use it with another codec, the complex algorithm must be re-run.

[ABV13] minimizes the degradation introduced by the speech codec by optimizing parameters of an M-FSK modem. Compared to [LSF08], this approach also takes into consideration a degradation caused by a discontinuous transmission. Optimal parameters are found during manual experimentation. This solution is not suitable for voice transmission, because it focuses on minimizing the error rate at a transfer rate too low to carry voice. A 16-FSK modem achieved a transfer rate of 800 bps while keeping a bit error rate below 10^{-3}.

[KVK03] describes a modem that modulates data by changing basic characteristics of a human speech. This work only illustrates modulation process, but provides no implementation details. [KANVK04] uses this modem for real time data transmission over a GSM voice channel. A throughput of three kbps has been achieved with a 2.9 percent byte error rate.

2.3 Modulation Software

Although a number of software tools implementing a modulation exist, it is possible to implement it from scratch. One option is to use GNU Octave [Eat], which is a high-level interpreted language. It is primarily intended for numerical computations and is widely used for signal processing. For achieving higher performance, low-level programming languages like C can be used.

2.3.1 Python QAM modem

[Pfü] is a python implementation of QAM designed for educational purposes. Its advantage is a simple and open-source code with a thorough documentation. Its disadvantages are poor performance and a zero tolerance to signal impairments.

2.3.2 J-QAM

J-QAM [Old10] is a proprietary QAM modem implementation with only binaries and header files available. It acquires speeds up to 300 kbps using QAM16 or QAM64. It also has an error correction feature.

This modem is able to stream live audio and video over any analog channel. J-QAM achieves a data rate of 84 kbps. In a long distance test of over one km, with air as a traffic medium, the data rate was 29 kbps.

A big disadvantage of this software is that it is hard to adapt for usage in mobile communication. Due to its closed source code, it is also impossible to adjust it for specific use case or port it to other platforms.

Furthermore, only some features are available in demo versions, resulting in a bandwidth of only 15 kbps. Finally, J-QAM does not operate in a narrow voice

band, but in a much wider frequency range and due to that achieves a higher bandwidth.

2.3.3 FDMDV Modem

A Frequency Division Multiplexed Digital Voice (FDMDV) Modem [Row12] is an open-source modem based on frequency division multiplexing. The modem is implemented in both C and Octave. The Octave version is meant for rapid prototyping, while the C version provides real time performance.

By default, FDMDV modem has 14 carriers, covering a frequency range of 600-1800 Hz. QPSK is used in each carrier generating 50 symbols per second. The modem achieves a maximum transfer rate of 1400 bps (14 carriers * 50 symbols * 2 bits per symbol). Since the modem uses only a part of a voice band, the bit rate can potentially be increased. The receiver side has a mechanism to deal with frequency offsets and certain signal impairments. Figure 2.1 illustrates signal impairments in a noisy channel.

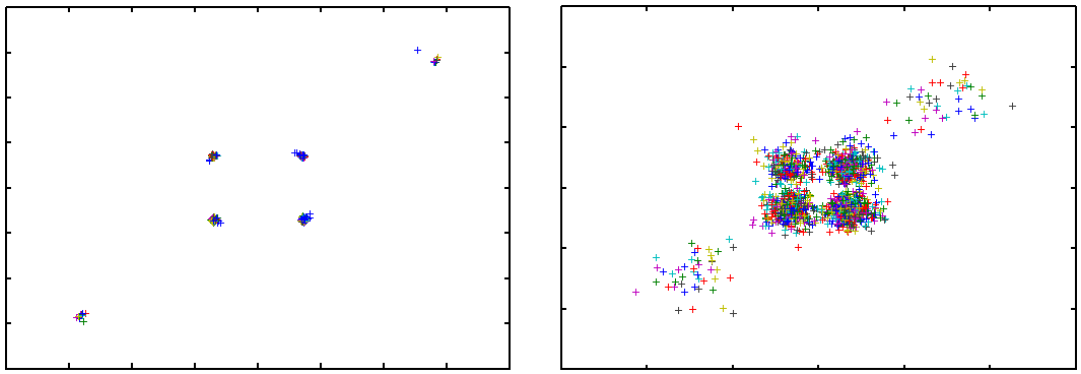

Figure 2.1: Effects of a noisy channel on a modulated signal. On the left is a scatter diagram of symbols modulated onto carrier signals. All 14 carriers are plotted on top of each other. Two separate dots on a diagonal represent a clock signal of the modem and have a slightly higher energy than the other symbols. On the right are the same symbols sent over a noisy channel [Row12].

The FDMDV modem has a number of advantages over alternative solutions. Firstly, it provides higher data transfer rates than the python implementation. Secondly, it is open to configuration, as opposed to the proprietary JQAM. For these reasons, we used this implementation in our system.

Chapter 3

System Overview

In this chapter, we describe a system that achieves a secure data transmission over a mobile voice channel with a further goal to provide secure voice transmission. Here we describe each component of the system in detail and discuss the issues that we encountered while building the system. Figure 3.1 gives an overview of the data flow during a usual mobile call and Figure 3.2 the data flow in our system.

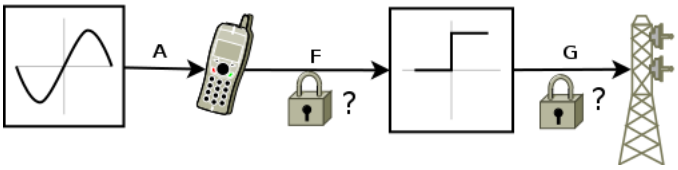

Figure 3.1: The data flow during a mobile call. A voice is captured by a mobile phone (A), transformed into digital data with a possible encryption (F) and sent over a mobile network (G).

3.1 Materials

3.1.1 Mobile Phones

Mobile phones used in this experiment are two Samsung Galaxy SII phones (different OS versions), Nokia N950, Nokia XpressMusic 5530 and Sony Erycsson Xperia Pro. All of these models have ports for 3.5 mm 4-conductor TRRS phone connectors (hands-free headset connector).

N950 and Xperia Pro both have an option to enable noise suppression, whereas XpressMusic and both Galaxy SII phones do not. We have upgraded an operation system of one of the Galaxy SII phones to enable noise suppression option.

In the course of experimentation, we have discovered that the XpressMusic would add background noise to the signal and interpret some input signal as "end call" command. For this reason, we did not use XpressMusic afterwards. In most

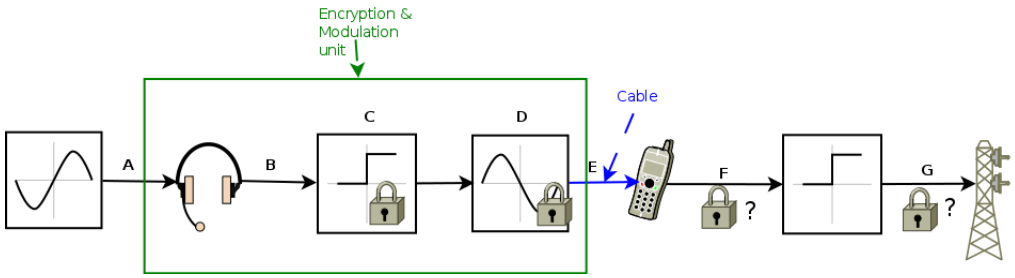

Figure 3.2: A secure voice channel scheme. To enforce a client-side encryption of a voice channel, a microphone captures the voice (A) before it reaches the mobile phone. The voice signal is then redirected into an encryption & modulation unit. An audio codec processes the analog voice signal into digital data (B). Digital data are then encrypted (C). Encrypted digital data are modulated into an analog signal (D). The data-carrying analog signal is sent to a mobile phone (E). Steps (F) and (G) follow as in a usual mobile call. In case of a digital data transmission, the voice recording and encoding steps are omitted and the process follows steps C-D-E-F-G.

of the experiments, we used a Galaxy SII as a sending device and either N950 or Xperia Pro as a receiver.

3.1.2 Encryption & Modulation Unit

In our prototype, an encryption and modulation unit is a Dell Latitude E5410 notebook computer. It has a 3.5 mm headphones output (headset-out) and a 3.5 mm microphone input (microphone-in) ports. It also has an internal microphone. The computer runs Debian GNU/Linux distribution with kernel version 3.11-2.

3.1.3 Custom Headset

To connect a mobile phone via its headset port to a microphone-in and headset-out ports of a computer, we modified a standard hands-free headset. See Figure 3.3. A typical headset cable usually provides two channels for an audio output from a mobile phone (left and right headphone), one auxiliary input for a microphone and a grounding pin. In the modified cable, an audio output goes into a 3.5 mm connector on the other end of the cable. Another 3.5 mm connector in connected to the auxiliary pin of the headset connector instead of a microphone.

Although all four phones have a headset port and recognize both headphones and a microphone with their stock headset, only Galaxy SII recognized the microphone on our custom headset.

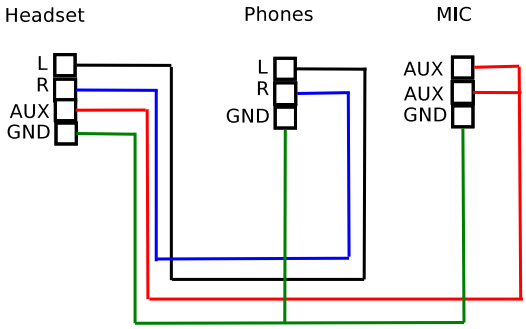

Figure 3.3: A modified headset cable which separates a 4-conductor headset connector into a headphone output and a microphone input connectors.

3.2 Input Processing

In our experiments, we used three types of input data: a plain text, an encrypted text and a pre-recorded voice signal. To ensure that the transmission link is correctly setup, we first used a plain text as an input. If the text was recognizable on the receiver end, the encrypted text and encoded audio files were sent.

Voice signal is pre-recorded as one 16-bit channel of PCM encoded audio data with a sample rate of 8000 Hz. Audio data are then compressed using a speech codec codec2 with a bit rate parameter set to 1200.

An encryption and decryption utility ccrypt was used for encryption. Ccrypt's prominent advantage over other alternatives (such as bcrypt and Open-SSL) is its capability to decrypt corrupt files. Alternatively, a Caesar cipher was used for byte-by-byte encryption.

This section covers steps A, B and C in Figure 3.2.

3.3 Transmitter

3.3.1 Modulation

The next step in our pipeline after input processing is signal modulation (step D in Figure 3.2). Based on an analysis of different modulation methods (section 1.1) and available software solutions (section 2.3), we used a modulation implementation in a FDMDV modem.

By default, the FDMDV modem uses a narrower frequency band of a carrier signal than that of human voice. Default settings place 14 carrier signals around a center frequency of 1500 Hz with 75 Hz separation between carriers. This results in a modulated signal in a frequency band of 900-2050 Hz and a data transmission rate of 1400 bps.

We modified the source code of the modem to use 20 carriers. The resulted modulated signal was in a frequency band of 700-2300 Hz. The data transfer rate of modified modulator was 2000 bps. For comparison of modulated signals, refer to Figure 3.4.

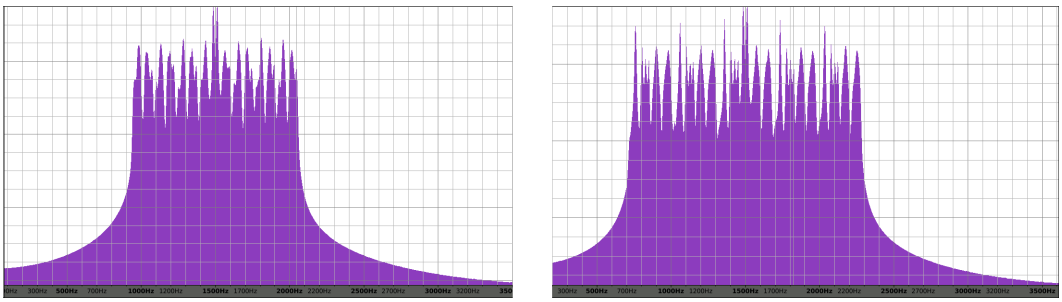

Figure 3.4: On the left is a plot spectrum of a signal created by an FDMDV modem with default settings. 14 data signals are multiplexed into one signal with frequency band of 900-2050 Hz. Two noticeably higher peaks in the center represent a clock signal of the modem, which has higher energy than data signals. Seven data signals are on each side of this center. On the right, the FDMDV modem multiplexed ten data signals onto each side of the center frequency. The multiplexed signal is in a frequency range of 700-2300 Hz.

We could have increased the data rate even more by filling the voice band with carriers and decreasing distance between them. The bottleneck in our experiments was, however, not the throughput of the channel, but a high error rate of the connection caused by channel noise and signal impairments.

3.3.2 Noise Suppression

After the modem has modulated data into a sound signal, we redirect in into the phone. At this point, the mobile phone can perform noise suppression. Although there was no noticeable effect on the input signal, the experiments showed that the error rate of transmission was lower up to ten percent with noise suppression enabled on the transmitting phone.

3.4 Receiver

In order to recover initial data from a signal that came over mobile voice channel, a process pictured in Figure 3.2 has to be performed in a reversed order. The incoming signal is passed from the phone through the modified headset and into the computer. Then the computer performs signal demodulation and data decryption. In case of a voice transmission, digital data represent an encoded sound file, which needs to be decoded and played back.

3.4.1 Recording Sound from Mobile Phone

The signal that has originated from a sending side has passed through a mobile voice channel into the receiving mobile phone and is redirected via headphones output into the computer. We discovered two types of signal impairments here.

In our experiments, a constant white noise would appear in the recorded signal, even if no device was attached to the microphone-in port of the computer. After some troubleshooting, two separate problems were pinpointed. A GNU/Linux kernel bug [lnx14] was a cause to some of the noise and was fixed by a patch.

Another cause of the noise was introduced by the internal microphone which happened to pick up noises made by the notebook itself [Tec14]. The noise would disappear after we unplugged notebook's charger and all peripheral devices.

3.4.2 Intermodulation Distortion

Intermodulation distortion is a signal integrity issue that arises when multiple signals are sent over a single transmission channel at the same time. The separate signals will mix or multiply with each other. The product of this mixing is called a signal harmonic. Harmonics are quieter than the original signal, but will mix and multiply again with the original signal and other harmonics. This creates a distinct pattern in a plot spectrum of the received signal (refer to Figure 3.5).

Intermodulation distortion is a major issue in systems where one medium is used by both transmitter and receiver. Examples of such systems include cellular base stations, duplex radio and wireless systems and satellite systems.

After we canceled out or prevented the noise caused by computer software and hardware, the error rates of most transmissions were still too high for us to understand the originating text or sound. Although the received modulated signal sounded like the original, the plot spectrums of two signals were different (refer to Figure 3.6).

The pattern in plot spectrum diagrams remained the same with 14- and 20-carrier signals. Further research showed that the same kind of distortion appears in a usual mobile conversation. This signal impairment looks like it was caused by intermodulation. A difference of frequencies between human voice and human voice sent over mobile channel can be seen in Figure 3.7.

3.4.3 Demodulation

After recording the modulated signal from a mobile phone, we pass it to the demodulator. The output of the demodulator is a stream of bits. We noted that this stream differs from original data.

Firstly, the data stream differs as a result of the demodulator trying to extract data from silence from the beginning of the input signal. Also, the beginning of the

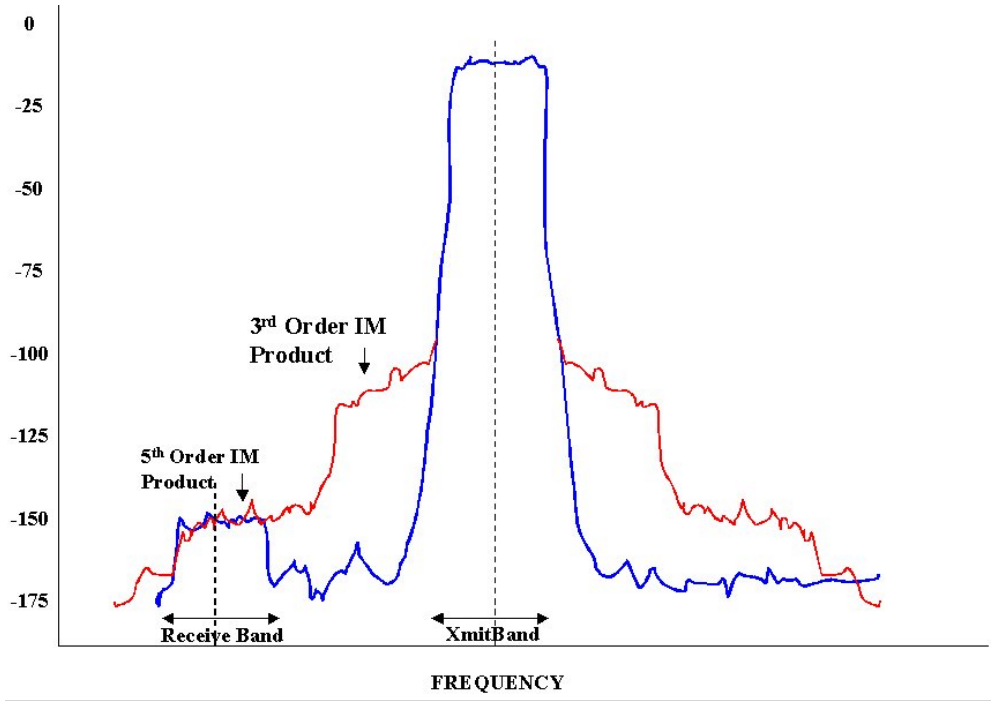

Figure 3.5: Intermodulation Distortion. 3rd and 5th order harmonics of transmitting signal create a distinctive noise pattern in a plot spectrum of the signal. In a mobile voice channel, harmonics of the transmitting signal overlap with the receiving signal. With voice, this creates insignificant distortion, with modulated data, the noise causes issues when demodulating the signal [fE].

data stream can be scrambled due to the calibration process of the demodulator.

Secondly, the demodulated data will have some incorrect bits as a result of mobile voice channel distortion. The negative effect of the mobile channel was heavier than we expected, due to the intermodulation distortion.

Finally, the amount of data will be bigger, because the demodulator will try to extract data from the silence at the end of the signal.

3.5 Post-Processing

The goal of a post-processing step is to restore original data from the stream of bits that comes from the demodulator output. In case of a voice transmission, demodulated data represent an audio file encoded by speech codec. The data has to be decoded and can be played back after that. It is not significant that the file has more data at the beginning and at the end than the original - it will sound as a short sequence of noise. The errors throughout the file are, on the other hand, an issue for the decoder and the player. The higher the error rate gets, the harder it will be to comprehend the extracted voice.

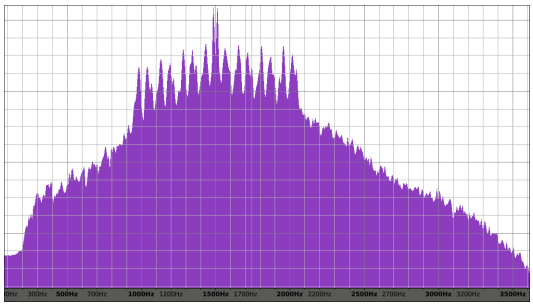

Figure 3.6: Plot spectrum of a signal received from mobile voice channel. 14 multiplex signal created by the FDMDV modem is noticeably distorted. The received signal contains noise on frequencies that were empty on the transmitting side. For comparison with the original signal, refer to Figure 3.4

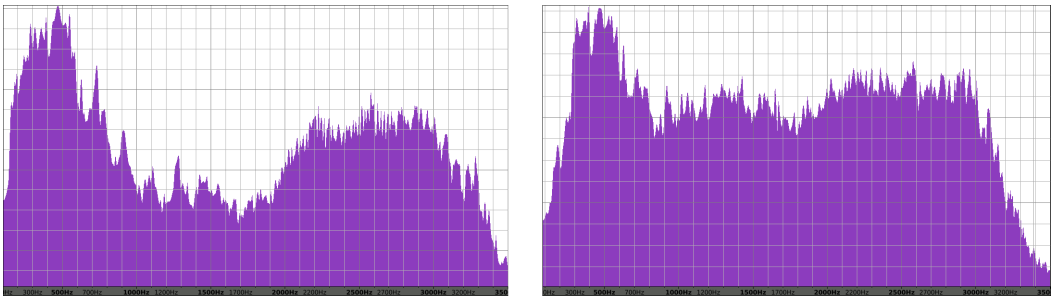

Figure 3.7: Mobile voice channel distortion of human voice. On the left is a plot spectrum of a human voice. On the right is a plot spectrum of a human voice received from mobile voice channel.

Alternatively, if the original data were an encrypted text file, it has to be decrypted. Since we know that the demodulated data is corrupted, we instruct the decryption utility to ignore mismatched keys and attempt the decryption.

Chapter 4

Experiments and Results

To test our system, we performed a number of experiments with different phones and modem parameters at different times of day and night. Since the FDMDV modem provided a transfer rate high enough for a real time voice data transmission (1400 bps transfer rate to carry a sound encoded by codec2 with data rate of 1200 bps), we were mostly interested in lowering the amount of errors in the transmission. Altogether we performed 83 documented transmission attempts using different types of data (see Appendix A for details).

4.1 Measuring the Error Rate

To measure the quality of the data transmission in a particular experiment, we compare the source file and the received file. We report the error rate as a fraction of scrambled bytes in the file (see script in Appendix B).

The received file, however, needs to be preliminary aligned. The reason is that we cannot automatically find the offset of the message in the received data stream. Also, the beginning of the data stream can be corrupted due to demodulator calibration.

4.2 Transmitting Different Types of Data

In order to asses the system performance, we performed a set of experiments with different types of data including plain text, encrypted text and voice data. In each experiment, we first fixed the parameters of the modem and tried to first transmit plain text data. If the transmission error rate was higher than 90 percent, we considered the experiment unsuccessful. In this case, we moved on to the next set of parameters. Otherwise, we repeated the same experiment with encrypted text and voice data. Figure 4.1 illustrates the distribution of observed error rates for each type of data.

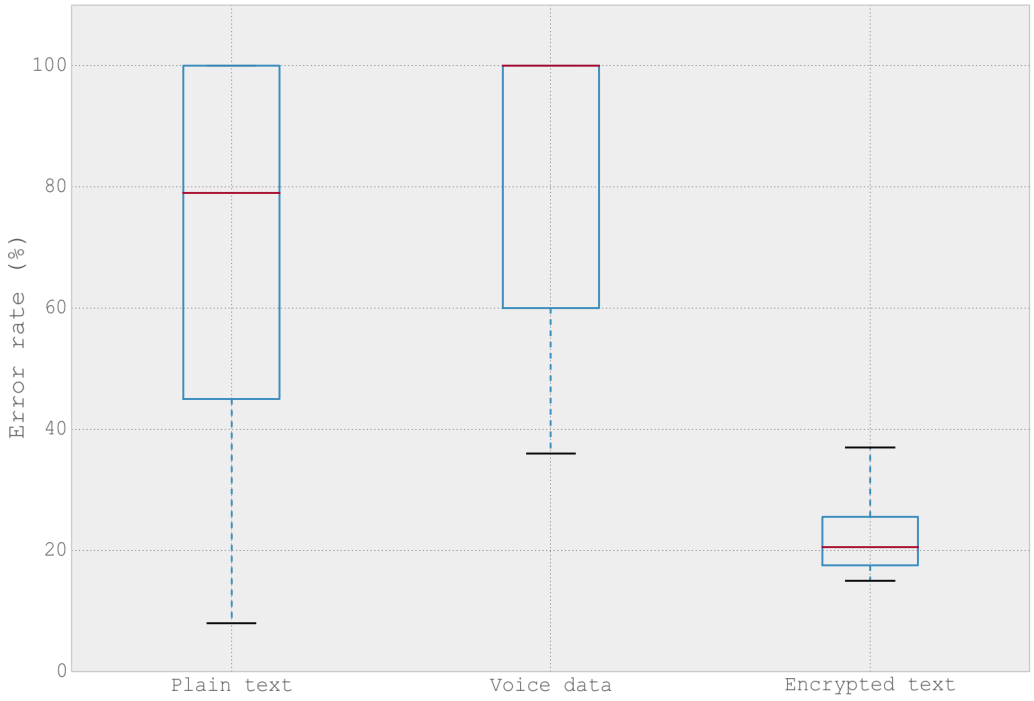

Figure 4.1: The distribution of error rates during transmission of different types of data.

We have made a total of 49 attempts with plain text, 15 of them failed. The median error rate was about 80 percent and the lowest error rate was eight percent. The high number of failed experiments is explained by our testing methodology. With plain text transmission, we tested new parameters of the modem and new phones. With an error rate below 20 percent, the text was readable, although it contained scrambled letters.

We have made eight attempts of encrypted text transmissions. Comparing the encrypted source file and the received file byte-to-byte, we found out that the median error rate was about 20 percent, i.e. every fifth byte was scrambled. Since the ccrypt utility uses a fixed block size of 16 bytes, it failed to recover the message. The resulted text was unreadable, except for some parts of sentences. With a byte-wise encryption, such as a Caesar cipher, only the scrambled bytes are not decrypted, producing similar results to the plain text transmission. The lowest error rate was 15 percent. A reason for low error rates is that we performed experiments with encrypted data only when previous experiments showed low error rates.

We have made 22 attempts of voice data transmission over mobile voice channel. In 16 of them, the error rate was higher than 90 percent. The lowest error rate was 36 percent. When the received data were played back, a voice intonation and pauses in the speech were clearly distinguishable. However, single words were

not comprehendable. A possible reason for high error rates is that the voice data file is longer than other an encrypted or plain text file.

4.3 Network Factor

In the course of experiments, we noticed that the error rate of the transmissions was not always affected by the parameters of the modem. The transmission would fail if the mobile network changed the signal too much, regardless of how many carrier frequencies the signal had or in which frequency band the signal was. We assumed that the reason is a busy mobile network.

To confirm our suspicions, we compared the results of tests done during working hours (8 AM - 9 PM) to the results of tests done at night and during weekends. We selected 35 test cases, where the modem parameters, source file and phones where the same. The only factor that changed was the time of the experiment. See Figure 4.2 for the comparison.

The error rate varied much more during working hours when the network is busy. The comparison also showed that we achieved the best results during weekends or nights. All of failed experiments, where error rate was higher than 90 percent, took place during working hours.

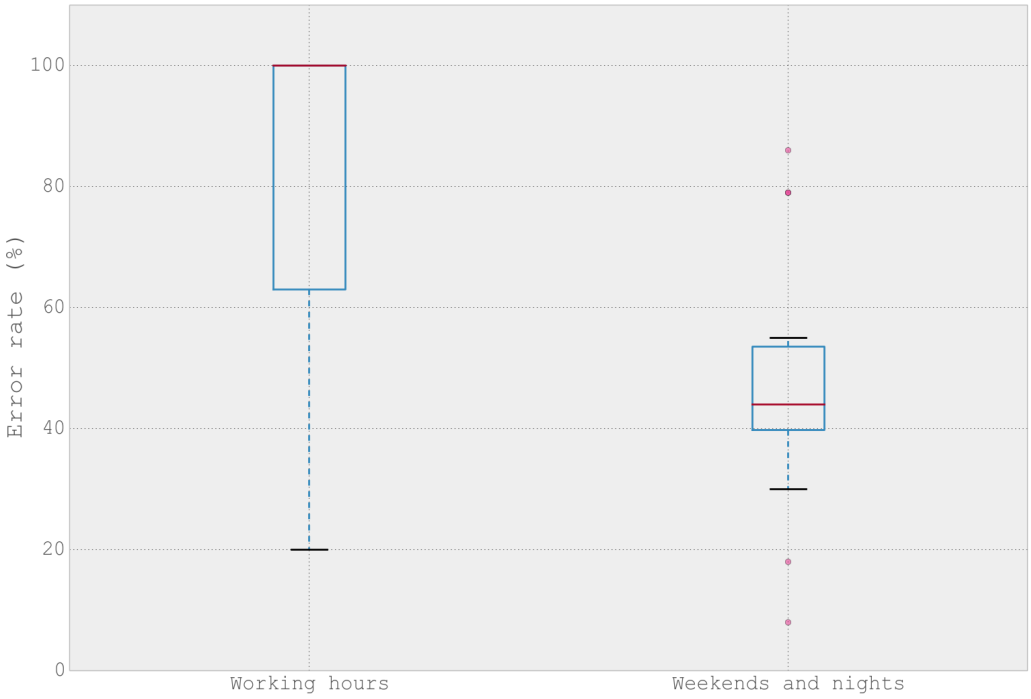

Figure 4.2: The error rate distribution of data transmission during working hours compared to weekends and nights.

4.4 Future Work

There are numerous ways to decrease the error rate of transmission that did not fit the scope of this thesis.

Firstly, it is possible to use a media broadcast protocol (such as RTP or raw UDP), similar to those used in Voice over IP. In this way, the data stream is split into packets before the transmission. This approach increases redundancy of data and decreases the error rate.

Secondly, we can implement a two way transmission that would allow the receiver to provide a real-time feedback of the transmission to the transmitter. This way, the transmitter can repeat the transmission of lost data or change parameters of the transmission and adjust to fluctuations in network quality.

Finally, we can use another modem instead of the FDMDV or rewrite the code of FDMDV to better suit our needs. The improved modem would cope better with signal impairments caused by the mobile network.

4.5 Discussion

One of the biggest issues during our experiments was the instability of the mobile network. It was hard to troubleshoot any problems in the system prototype because the results would vary in identical experiments.

Mobile network was also the only part in the system that we had no control over and therefore could not change it. If a problem occurred with mobile phones, we would retry the experiment with different device. When the suspicion fell onto our custom-made headset cable, another one was assembled. We could not, however, change the mobile voice channel or get any feedback of what was happening to the data over the air.

Nonetheless, it was a good experience to finally get the system to work and recognize original data in the received stream of bits.

Conclusion

In this paper, we have described a system that performs data transmission over a mobile voice channel using a FDMDV software modem, a computer and a modified headset cable. Using this system, our goal was to achieve secure data transmission on a mobile voice channel. In addition, we experimented on passing voice over the established secure channel.

We provided background information on modulation technologies as well as on mobile networks and voice recording. We then described issues that we encountered in our experiments. The problems included intermodulation distortion in the mobile channel that the demodulator could not cancel out.

As a result, our system established a secure data connection with transfer speeds up to 2000 bps and a median error rate of 21 percent. Because of the high error rate, the channel we provided was not reliable enough to carry a voice signal.

References

[ABV13] Bechir Taleb Ali, Geneviève Baudoin, and Olivier Venard. Data transmission over mobile voice channel based on M-FSK modulation. In *WCNC*, pages 4416–4421. IEEE, 2013.

[Boc12] Matthias Bock. Frequency-division multiplexing (FDM): The spectrum of each input signal is shifted to a distinct frequency range. `https://www.commons.wikimedia.org`, 2012.

[CM] Cellcrypt-Mobile. Cellcrypt Voice Security Solutions. `http://www.cellcrypt.com/cellcrypt-mobile`.

[Com] Nabishi Secure Communications. Secure Voice Compact GSM Phone. `http://ww2.nabishi.com/nab-admin/index.php/secure-gsm-mobiles/secure-voice-gsm/`.

[com14] Wikimedia Commons. `https://www.commons.wikimedia.org`, 2014.

[Cry] GSMK Cryptophone. Trustworthy Voice and Message Encryption. `http://www.cryptophone.de/`.

[Des12] Lou Frenzel | Electronic Design. Understanding Modern Digital Modulation Techniques. `http://electronicdesign.com/communications/understanding-modern-digital-modulation-techniques`, 2012.

[Eat] John W. Eaton. GNU Octave. `https://www.gnu.org/software/octave/`.

[fE] Arlon Materials for Electronics. Frequently Asked Questions About Microwave Materials. `http://www.arlon-med.com/index.cfm?fuseaction=content.faq&faqTypeID=40013&faqCatID=40045`.

[FJHT14] Andreas Bakke Foss, Per Anders Johansen, and Fredrik Hager-Thoresen. Secret surveillance of Norway's leaders

detected. http://www.aftenposten.no/nyheter/iriks/Secret-surveillance-of-Norways-leaders-detected-7825278.html, 2014.

[KANVK04] N.N. Katugampala, K.T. Al-Naimi, S. Villette, and A.M. Kondoz. Real time data transmission over GSM voice channel for secure voice and data applications. In *Secure Mobile Communications Forum: Exploring the Technical Challenges in Secure GSM and WLAN, 2004. The 2nd IEE (Ref. No. 2004/10660)*, pages 7/1–7/4, Sept 2004.

[kon12] Eesti Vabariigi Põhiseadus - Kommenteeritud Väljaanne, Paragrahv 43. http://www.pohiseadus.ee/pg-43, 2012.

[KVK03] N. Kaiugampala, S. Villette, and A.M. Kondoz. Secure voice over GSM and other low bit rate systems. In *Secure GSM and Beyond: End to End Security for Mobile Communications, IEE Seminar on (Digest No. 2003/10059)*, pages 3/1–3/4, Feb 2003.

[lnx14] Headset/headphones background noise. http://xps13-9333.appspot.com/#background_noise, 2014.

[LSF08] Christoph K. LaDue, Vitaliy V. Sapozhnykov, and Kurt S. Fienberg. A Data Modem for GSM Voice Channel. *IEEE T. Vehicular Technology*, 57(4):2205–2218, 2008.

[Mes03] R. Meston. Sorting Through GSM Codecs: A tutorial. *Irvine*, 2003.

[O'B09] Kevin J. O'Brien. Cellphone Encryption Code Is Divulged. http://www.nytimes.com/2009/12/29/technology/29hack.html?pagewanted=all, 2009.

[Old10] Jonti Olds. JQAM. A QAM soundcard modem. http://www.dxzone.com/cgi-bin/dir/jump2.cgi?ID=18491, 2010.

[Pfü] Elvis Pfützenreuter. QAM modem. https://epx.com.br/artigos/qam_tx.php.

[RE] Radio-Electronics.com. Resources and analysis for electronics engineers. http://www.radio-electronics.com/.

[Row11] David Rowe. Codec 2. http://www.rowetel.com/blog/?page_id=452, 2011.

[Row12] David Rowe. FDMDV Modem. http://www.rowetel.com/blog/?page_id=2458, 2012.

[RWO98] Siegmund Redl, Matthias Weber, and Malcolm W. Oliphant. *GSM and Personal Communications Handbook*. Artech House, Norwood, Massachusetts, May 1998.

[SB13] Philip Sherwell and Louise Barnett. Barack Obama 'approved tapping Angela Merkel's phone 3 years ago'. `http://fw.to/wqz2vVW`, 2013.

[Sel00] Peter Selinger. Ccrypt. `http://ccrypt.sourceforge.net/`, 2000.

[ssl99] OpenSSL Cryptography and SSL/TLS Toolkit. `https://www.openssl.org/`, 1999.

[Sys14] Open Whisper Systems. RedPhone :: Private Calls. `https://play.google.com/store/apps/details?id=org.thoughtcrime.redphone&hl=en`, 2014.

[Tec14] Dell TechCenter. OS and Applications. `http://en.community.dell.com/techcenter/os-applications/f/4613/p/19541816/20635745#20635745`, 2014.

[Tri] Tripleton. The Absolute in Secure GSM Communications. `http://www.tripleton.com`.

[uS06] Wikipedia user Splash. BPSK and 16-QAM symbol constellations. `https://www.commons.wikimedia.org`, 2006.

[xosc] The xiph open source comminity. Speex: A Free Codec For Free Speech. `http://www.speex.org/`.

All online sources were last accessed on 29.12.2014.

Appendices

Appendix A. Experiment protocol

```
Abbreviations:
xpress - Nokia XpressMusic 5530
n950 - Nokia N950
xperia - Sony Erycsson Xperia Pro
galaxy - Samsung Galaxy SII (CyanogenMod 9, Android 4.0)
galaxy_new - Samsung Galaxy SII (CyanogenMod 11, Android 4.4.4)
samsung - Samsung Galaxy SII (Android 2.3)
```

ID	input data	speech codec	error rate	dist betw carriers (Hz)	frequency band (Hz)	No of carriers	transfer rate (bps)	phone1	phone2	phone1 noise surpression	phone2 noise surpression	weekday	time	notes
1	sound	speex	>90%	75	900-2050	14	1410	xpress	n950	-	on	sun	05:00:00 PM	sound unrecognizable
2	plain text	-	90%	75	900-2050	14	1400	xpress	n950	-	on	sun	05:00:00 PM	
3	plain text	-	>90%	75	900-2050	14	1400	xpress	xperia	-	on	sun	06:00:00 PM	
4	plain text	-	50%	75	900-2050	14	1400	galaxy	xperia	-	on	sun	06:00:00 PM	
5	plain text	-	>90%	75	900-2050	14	1400	galaxy	xperia	-	on	mon	02:00:00 PM	
6	plain text	-	>90%	75	900-2050	14	1400	galaxy	xperia	-	off	mon	06:00:00 PM	
7	plain text	-	>90%	75	900-2050	14	1400	galaxy	xperia	-	off	mon	06:30:00 PM	
8	plain text	-	>90%	75	900-2050	14	1400	galaxy	xperia	-	off	mon	06:40:00 PM	
9	plain text	-	>90%	75	900-2050	14	1400	galaxy	xperia	-	off	mon	06:50:00 PM	
10	plain text	-	>90%	75	900-2050	14	1400	galaxy	xperia	-	off	mon	07:10:00 PM	
11	plain text	-	>90%	75	900-2050	14	1400	galaxy	xperia	-	off	thu	07:50:00 PM	
12	plain text	-	>90%	75	900-2050	14	1400	galaxy	xperia	-	off	thu	08:00:00 PM	
13	plain text	-	>90%	75	900-2050	14	1400	galaxy	xperia	-	off	fri	01:00:00 PM	
14	-	-	-	-	-	-	-	xpress	galaxy	-	-	thu	08:00:00 PM	xpress interprets some input as “end call” signal
15	plain text	-	30%	75	900-2050	14	1400	galaxy	xperia	-	off	thu	09:00:00 PM	
16	-	-	-	-	-	-	-	xperia	galaxy	-	-	sun	07:00:00 PM	xperia wont recognize microphone of the modified headset cable
17	plain text	-	55%	75	900-2050	14	1400	galaxy	xperia	-	off	sun	08:10:00 PM	
18	plain text	-	18%	75	900-2050	14	1400	galaxy	xperia	-	on	sun	08:15:00 PM	
19	plain text	-	8%	75	900-2050	14	1400	galaxy	xperia	-	on	sun	08:30:00 PM	
20	encrypted text	-	16%	75	900-2050	14	1400	galaxy	xperia	-	on	sun	08:35:00 PM	text partially recognizable
21	plain text	-	45%	75	900-2050	14	1400	galaxy	xperia	-	on	sun	11:00:00 PM	

22	sound	speex	>90%	75	900-2050	14	1400	galaxy	xperia	-	on	sun	09:15:00 PM	sound unrecognizable
23	sound	speex	>90%	75	900-2050	14	1400	galaxy	xperia	-	on	sun	10:50:00 PM	sound unrecognizable
24	plain text	-	60%	75	900-2050	14	1400	galaxy_new	n950	off	off	mon	01:00:00 PM	
25	sound	codec2	>90%	75	900-2050	14	1400	galaxy_new	n950	off	off	mon	01:10:00 PM	sound unrecognizable
26	sound	raw	>90%	75	900-2050	14	1400	galaxy_new	n950	off	off	mon	01:20:00 PM	sound unrecognizable
27	plain text	-	45%	75	900-2050	14	1400	galaxy_new	n950	on	on	mon	01:30:00 PM	
28	sound	codec2	>90%	75	900-2050	14	1400	galaxy_new	n950	on	on	mon	01:40:00 PM	sound unrecognizable
29	sound	raw	-	75	900-2050	14	1400	galaxy_new	n950	on	on	mon	01:50:00 PM	sound unrecognizable
30	plain text	-	50%	75	900-2050	14	1400	galaxy_new	xperia	off	off	thu	01:45:00 PM	
31	sound	codec2	60%	75	900-2050	14	1400	galaxy_new	xperia	off	off	thu	01:45:00 PM	sound unrecognizable
32	sound	raw	-	75	900-2050	14	1400	galaxy_new	xperia	off	off	thu	01:45:00 PM	sound unrecognizable
33	plain text	-	>90%	75	900-2050	14	1400	galaxy_new	xperia	on	on	thu	01:45:00 PM	
34	sound	codec2	>90%	75	900-2050	14	1400	galaxy_new	xperia	on	on	thu	01:45:00 PM	sound unrecognizable
35	plain text	-	50%	75	900-2050	14	1400	galaxy_new	xperia	on	on	thu	02:00:00 PM	
36	sound	codec2	-	75	900-2050	14	1400	galaxy_new	xperia	on	off	thu	02:15:00 PM	sound unrecognizable
37	sound	raw	-	75	900-2050	14	1400	galaxy_new	xperia	on	off	thu	01:45:00 PM	sound unrecognizable
38	plain text	-	>90%	75	900-2050	14	1400	galaxy_new	xperia	on	on	thu	02:00:00 PM	
39	plain text	-	40%	75	900-2050	14	1400	galaxy_new	xperia	on	on	mon	01:40:00 AM	
40	plain text	-	42%	75	900-2050	14	1400	galaxy_new	xperia	on	on	mon	03:00:00 AM	
41	plain text	-	39%	75	900-2050	14	1400	galaxy_new	xperia	on	on	mon	02:50:00 AM	
42	plain text	-	-	75	900-2050	14	1400	galaxy_new	xperia	on	on	mon	03:15:00 AM	less data received than was sent
43	plain text	-	55%	75	900-2050	14	1400	galaxy_new	xperia	on	on	tue	02:20:00 PM	
44	encrypted text	-	>90%	75	900-2050	14	1400	galaxy_new	xperia	on	on	tue	02:20:00 PM	unable to decrypt, dcrypt used
45	sound	codec2	44%	75	900-2050	14	1400	galaxy_new	xperia	on	on	tue	02:20:00 PM	sound unrecognizable

46	sound	raw	-	75	900-2050	14	1400	galaxy_new	xperia	on	on	tue	02:20:00 PM	sound unrecognizable
47	plain text	-	-	75	900-2050	14	1400	galaxy_new	xperia	on	on	mon	04:40:00 PM	less data received than was sent
48	plain text	-	39%	75	900-2050	14	1400	galaxy_new	xperia	on	on	mon	04:40:00 PM	
49	sound	codec2	50%	75	500-3000	28	2800	galaxy_new	xperia	on	on	wed	04:40:00 PM	sound unrecognizable
50	sound	codec2	50%	75	500-3000	28	2800	galaxy_new	xperia	on	on	wed	04:40:00 PM	sound unrecognizable
51	sound	codec2	>90%	75	500-3000	28	2800	galaxy_new	xperia	on	on	wed	04:40:00 PM	sound unrecognizable
52	plain text	-	-	75	500-3000	28	2800	galaxy_new	xperia	on	on	wed	05:20:00 PM	less data received than was sent
53	sound	codec2	84%	75	500-3000	28	2800	galaxy_new	xperia	on	on	wed	05:20:00 PM	sound unrecognizable
54	plain text		-	75	500-3000	28	2800	galaxy_new	xperia	on	on	wed	05:20:00 PM	
55	plain text	-	-	75	500-3000	28	2800	galaxy_new	xperia	on	on	thu	01:10:00 PM	less data received than was sent
56	sound	codec2	-	75	500-3000	28	2800	galaxy_new	xperia	on	on	thu	01:10:00 PM	less data received than was sent
57	plain text	-	>90%	10	1300-1600	14	1400	galaxy_new	xperia	on	on	thu	01:40:00 PM	
58	sound	codec2	>90%	10	1300-1600	14	1400	galaxy_new	xperia	on	on	thu	01:40:00 PM	
59	plain text	-	-	10	1300-1600	14	1400	galaxy_new	xperia	on	on	thu	01:40:00 PM	less data received than was sent
60	plain text	-	>90%	150	30-3000	14	1400	galaxy_new	xperia	on	on	thu	02:50:00 PM	
61	plain text	-	86%	75	900-2050	14	1400	galaxy_new	xperia	on	on	sat	05:30:00 PM	
62	encrypted text	-	>90%	75	900-2050	14	1400	galaxy_new	xperia	on	on	sat	05:30:00 PM	unable to decrypt, dcrypt used
63	plain text	-	79%	75	1600-2800	14	1400	galaxy_new	xperia	on	on	mon	02:30:00 AM	
64	plain text	-	-	75	1100-1900	8	780	galaxy_new	xperia	on	on	mon	03:30:00 AM	less data received than was sent
65	plain text	-	-	75	700-2300	20	2000	galaxy_new	xperia	on	on	mon	03:30:00 AM	less data received than was sent
66	plain text	-	20%	75	700-2300	20	2000	galaxy_new	xperia	on	on	mon	04:30:00 PM	
67	plain text	-	71%	75	500-3000	14	700	galaxy_new	xperia	on	on	sat	06:30:00 PM	bpsk used

68	plain text	-	33%	75	900-2050	14	1400	galaxy_new	samsung	on	-	sun	05:40:00 PM	
69	encrypted text	-	37%	75	900-2050	14	1400	galaxy_new	samsung	on	-	sun	05:45:00 PM	file decrypted, text unrecognizable
70	sound	codec2	36%	75	900-2050	14	1400	galaxy_new	samsung	on	-	sun	05:50:00 PM	sound unrecognizable
71	plain text	-	45%	75	900-2050	14	1400	galaxy_new	samsung	on	-	sun	05:55:00 PM	
72	encrypted text	-	33%	75	900-2050	14	1400	galaxy_new	samsung	on	-	sun	06:05:00 PM	text partially recognizable
73	sound	codec2	-	75	900-2050	14	1400	galaxy_new	samsung	on	-	sun	06:10:00 PM	sound unrecognizable
74	plain text	-	52%	75	900-2050	14	1400	galaxy_new	xperia	on	on	tue	11:00:00 PM	
75	plain text	-	79%	75	900-2050	14	1400	galaxy_new	xperia	on	on	tue	11:00:00 PM	
76	plain text	-	53%	75	900-2050	14	1400	galaxy_new	xperia	on	on	tue	11:00:00 PM	
77	plain text	-	42%	75	900-2050	14	1400	galaxy_new	xperia	on	on	tue	11:00:00 PM	
78	plain text	-	43%	75	900-2050	14	1400	galaxy_new	xperia	on	on	tue	11:00:00 PM	
79	encrypted text	-	19%	75	900-2050	14	1400	galaxy_new	xperia	on	on	wed	06:00:00 PM	
80	encrypted text	-	15%	75	900-2050	14	1400	galaxy_new	xperia	on	on	wed	06:00:00 PM	
81	encrypted text	-	23%	75	900-2050	14	1400	galaxy_new	xperia	on	on	wed	06:00:00 PM	
82	encrypted text	-	18%	75	900-2050	14	1400	galaxy_new	xperia	on	on	wed	06:00:00 PM	
83	encrypted text	-	22%	75	900-2050	14	1400	galaxy_new	xperia	on	on	wed	06:00:00 PM	

Appendix B. Python script for error rate estimation

```
import sys
f1 = open(sys.argv[1]).read()
f2 = open(sys.argv[2]).read()
same = sum(1 for a, b in zip(f1, f2) if a == b)
print 1 - float(same) / len(f1)
```

Non-exclusive licence to reproduce thesis and make thesis public

I, Maksim Lind (date of birth: 06.07.1992),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   "Secure Data Transmission over Mobile Voice Channel", supervised by Alexander Tkachenko.

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 07.01.2015