TECHNICAL UNIVERSITY OF CRETE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
TELECOMMUNICATIONS DIVISION

Information Steganography using Backscatter Radio

by

Athanasios Topalis

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DIPLOMA OF

ELECTRICAL AND COMPUTER ENGINEERING

August 2019

THESIS COMMITTEE

Professor Aggelos Bletsas, *Thesis Supervisor*
Professor George N. Karystinos
Associate Professor Michail G. Lagoudakis

Abstract

Recent advances in ambient, FM backscatter radio showed the feasibility of piggybacking an audio signal on top of a broadcast FM radio transmission. In conjunction with the weak transmission power inherent in backscatter radio systems, this work exploits FM ambient backscatter to achieve covert communication. Information in the form of plain text is converted to a wideband audio signal using Direct Sequence Spread Spectrum (DSSS) modulation, resulting in a noise-like audio signal undergoing Frequency Modulation (FM) before being backscattered to a conventional smart phone equipped with FM radio, acting as the receiver. A Linear Feedback Shift Register (LFSR), provided with a key completely determining its output, generates the symbol sequences that serve as spreading sequences in DSSS, thus adding a Private-Key Encryption layer to enhance the cryptosystem's robustness. A Repetition Code is utilized as a way of error correction to increase the performance. Besides simulations, a prototype was implemented to verify the feasibility of the proposed concept.
Thesis Supervisor: Professor Aggelos Bletsas

Acknowledgements

I would like to thank my supervisor, Professor Aggelos Bletsas, from the bottom of my heart for his guidance and understanding. You have shown me the way and encouraged me to continue my studies in our field. Thank you.

I was extremely lucky to work with Giorgos V., Michail Our., Konstantinos S., Vaggelis G., Vaggelis K. and complete this thesis in TUC Telecom Lab. I wish you guys the best and I will be waiting for you in Stockholm.

I truly wish I would never part ways with Giorgos V., Sotiris L. and Vaggelis T. You have been family to me and I will miss you the most.

Nikos T. and Alex P. have been there since day one. We have put each other in dilemmas and we share too many great night stories to be written.

Michail Our. has helped me change my perception about many things. You deserve everything you may achieve in your life. I want a Celtics jersey.

Giannis P. has become a great friend. I hope you join me in Sweden.

Nick K. and Fani K. were the first to work with and earn my own money. I am sure that you will become well-known musicians someday.

Mr. Ilias has been a very important person in my student life. The reason I became the man I am is because of the endless conversations we've had. Kyr Ilia, I will never forget you. Promise.

My uncle Giorgos is the reason I chose to become an electrical engineer and continue my studies in Sweden. Since I was young, he was that cool guy I always wanted to be like, so I decided to follow his path. Thanks Theio.

Giannis M., Giorgos S. and Dimitris T. have been with me through thick and thin. It is them who helped me shape my character back in high school. Nowadays, we could all play a role in "No Country for Old Friends".

Finally, my parents have supported me in each and every step I took. I love you guys. "Dado", I wish you were here.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Audio Steganography

Steganography is the practice of hiding information in such a way that no one, except the transmitter and the intended receiver, suspect the existence of a message.

Audio Steganography is a technique used to transmit hidden information by modifying an audio signal in an imperceptible manner. Audio signals have frequencies in the audio spectrum, a range of roughly 20 to 20,000 Hz, which corresponds to the lower and upper limits of human hearing. An audio *message* signal carrying the information is concealed in an audio *host* signal. The *host* signal (before steganography) and *stego* signal (after steganography) have the same characteristics.

There are various audio steganography methods as shown in Fig. 1.1 and [3]. In this work, the *host* signal is a radio-broadcast audio signal and its content is unknown to the receiver. Spread Spectrum method is implemented, as the resulting wideband channel is more resistant to both unintentional and intentional interference.

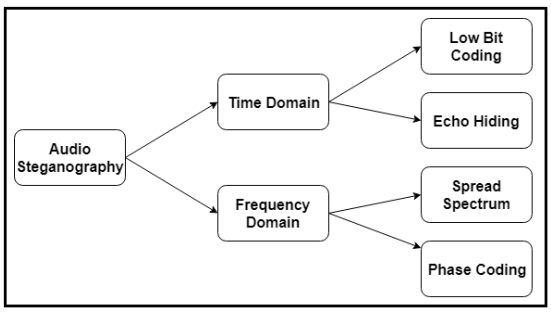

Figure 1.1: Audio Steganography Methods

1.2 LFSR Encryption

A Linear-Feedback Shift Register (LFSR) is a finite state machine which finds extensive use in cryptography [4]. The initial state of the LFSR is called the *seed* and the operation of the register is deterministic, meaning that the current register state is completely determined by its previous state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function (the bit positions that affect the next state, called *taps*), can produce a sequence of bits which appears random and which has a very long cycle. The tapped bits can be expressed as the powers of a polynomial's terms, such as $x^4 + x^3 + 1$, which means that the 4_{th} and 3_{rd} bit define the input bit after a register shift (cycle). The maximal length sequence for a given register size can be achieved only if the polynomial determining the taps is primitive. For example, a 4-bit LFSR tapped by a primitive feedback polynomial produces a $2^4 - 1 = 15$ bit sequence, which is the longest for this register size.
In this work, the LFSR will be used in a slightly different way, as the bits of the current register state will be referred to as the output bit sequence, meaning that the sequence length will be equal to the register length.

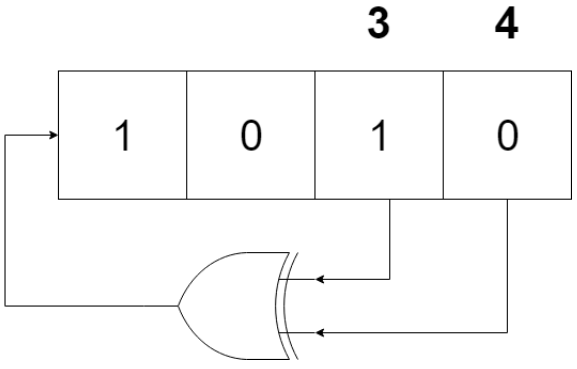

Figure 1.2: A 4-bit Fibonacci LFSR

In case of a 4-bit register shown in Fig. 1.2, the primitive feedback polynomial $x^4 + x^3 + 1$ is selected in order to achieve the maximum period. As shown in Fig.1.3, the sequence will repeat itself after $2^4 - 1 = 15$ cycles,

as the feedback polynomial secures that the register shall never reach the all-zeros state, which is excluded by the total possible binary combinations.

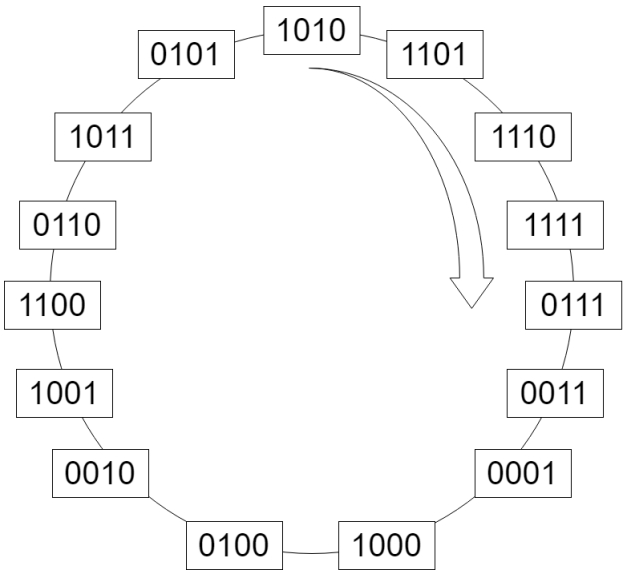

Figure 1.3: State cycle of a 4-bit LFSR

An LFSR-generated sequence, as explained above, can be used both as a spreading sequence to implement the Direct Sequence Spread Spectrum method [5] and as a means of encrypting the packet at the same time, because only the transmitter and the intended receiver will know the spreading sequences, if the key (consisting of the *seed* and *taps*) remains secret.

1.3 FM Ambient Backscatter Radio

FM Ambient Backscatter is a technology that uses existing FM radio signals to transmit data without a battery. Such a device (referred to as *tag*) is comprised of an antenna to pick up the radio signal broadcast by an FM station, as shown in Fig. 1.4. The tag is configured to backscatter the impinged signal to convey its own (audio) information, as shown in [1]. A smart phone equipped with FM radio can act as the receiver to record the demodulated audio signal.

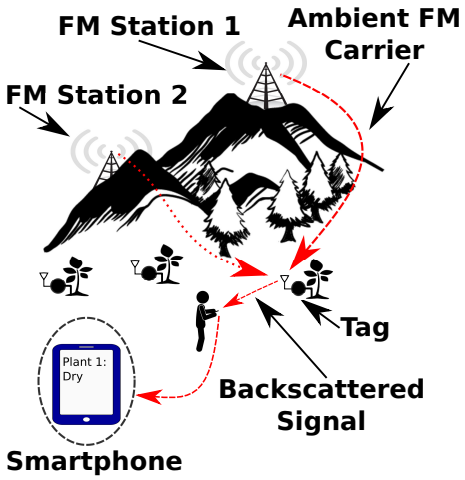

Figure 1.4: FM Ambient Backscatter [1]

Energy is only needed for an RF switch to alternate the termination of the antenna between two loads, thus modulating the impinged RF signal using backscatter radio principles [1].
In this thesis, the tag of [1] was implemented using a function generator as an FM modulator, whose input is the secret message to be transmitted converted to an audio signal $\mu(\tau)$.
The FM-modulated audio signal $\mu(\tau)$ driving the RF switch is given by:

$$x_{sw,FM}(t) = A_{sw} \cos\left(2\pi F_{sw} t + 2\pi k_{sw} \int_0^t \mu(\tau)\, d\tau \right),$$

where A_{sw} and F_{sw} is, respectively, the amplitude and fundamental frequency of a sine wave modulated by $\mu(\tau)$. The modulator's frequency sensitivity

$k_{sw} = \Delta f_{max}$ when $\max|\mu(\tau)| = 1$, so the audio signal $\mu(\tau)$ is put through an operational amplifier to control its maximum amplitude, in order to achieve a desirable deviation Δf_{max} .
The illuminating carrier is an FM-modulated audio signal $\phi(\tau)$ broadcast by an FM radio station, whose model is given by:

$$c_{st}(t) = A_{st}\ \cos\Big(\ 2\pi F_{st} t\ +\ 2\pi k_{st} \int_0^t \phi(\tau)\ d\tau\ \Big)\ ,$$

where A_{st} is a carrier's characteristic, F_{st} the carrier (station) center frequency and k_{st} is its modulator's frequency sensitivity.
When the carrier wave impinges on tag's antenna, a signal is backscattered:

$$y_{bs}(t) = \sqrt{\eta}\ c_{st}(t)\ x_{sw,FM}(t)\ ,$$

where η is the scattering efficiency.
The illuminating signal and the switching signal are mixed, resulting in the backscattered signal being the sum of the two FM signals, one at $F_{st} + F_{sw}$ and another one at $F_{st} - F_{sw}$. The backscattered signal $y_{bs}(t)$ at each of the two aforementioned center frequencies is also FM and can be described by the following equation:

$$\begin{aligned} y_{bs}(t) = &\ \frac{\sqrt{\eta} A_{sw} A_{st}}{2}\ \cos\Big(\ 2\pi\big(F_{st} + F_{sw}\big)\ t + \Phi_{st}(t)\ + \Phi_{tag}(t)\Big) + \\ &+ \frac{\sqrt{\eta} A_{sw} A_{st}}{2}\ \cos\Big(\ 2\pi\big(F_{st} - F_{sw}\big)\ t + \Phi_{st}(t) - \Phi_{tag}(t)\Big)\ , \end{aligned}$$

where $\Phi_{tag}(t) = 2\pi k_{sw} \int_0^t \mu(\tau)\ d\tau$ and $\Phi_{st}(t) = 2\pi k_{st} \int_0^t \phi(\tau)\ d\tau$.
Finally, the demodulated backscattered signal (for e.g., $F_{st} + F_{sw}$) can be expressed as:

$$y_{aud}(t) = \mu(t) + \phi(t) + n(t)\ ,$$

where $n(\tau)$ is the audio noise affecting the signal due to the RF signal cor-

ruption.

Making use of this conclusion, a baseband system model of embedding the audio-converted information to the station audio signal is described in the following section.

Chapter 2

Baseband System Model

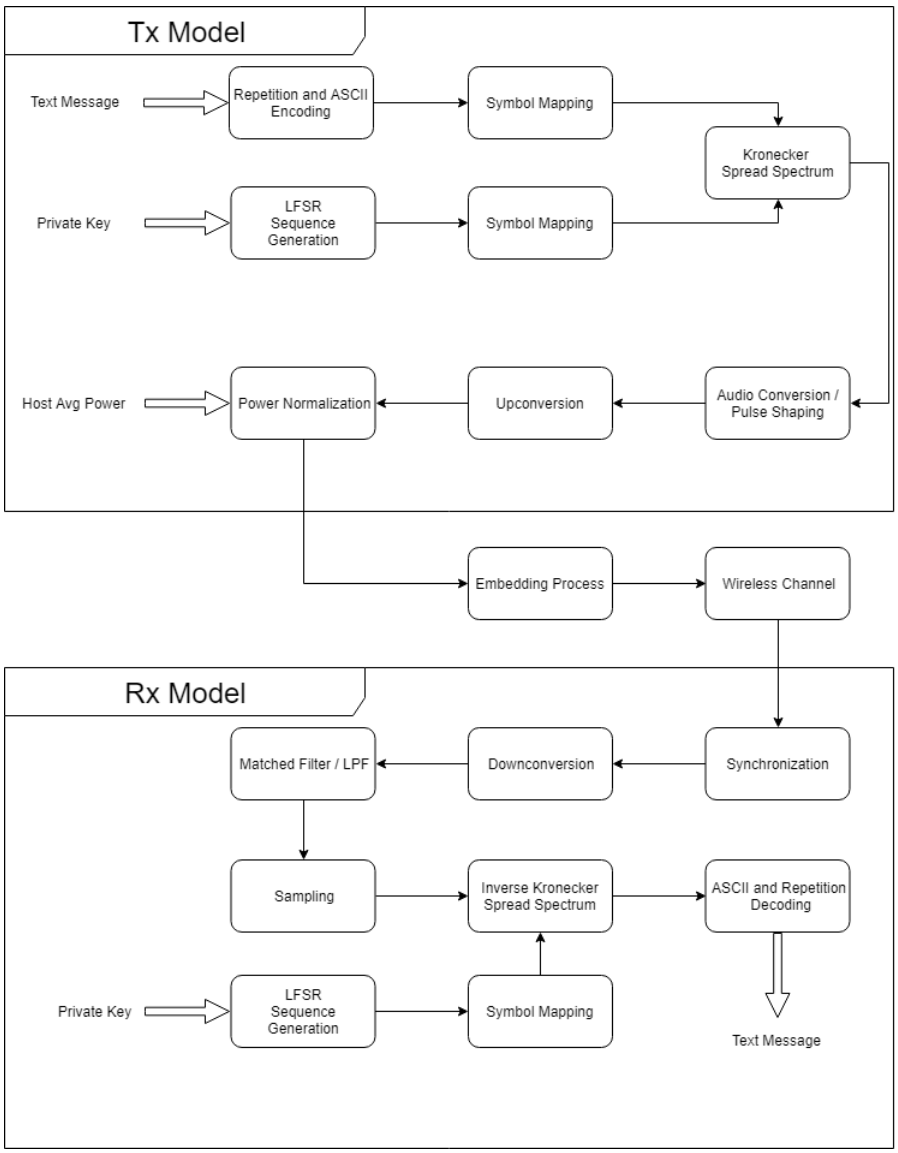

Figure 2.1: System Model Block Diagram

2.1 Transmitter Design

Character Buffer Construction and Repetition Encoding

Let $\mathbb{V} = \{C\}$ be the non-extended ASCII printable character set : $|\mathbb{V}| = 95$. Its cardinality is not 128 as expected, because the first 32 symbols in the ASCII table, which represent control characters, and the "DEL" character are excluded, as none of them are printable. A character $C_{sync} \in \mathbb{V}$ is used as the Sync Character. Assume a secret text message to be transmitted, consisting of N_{char} alpharithmetic characters:

$$\mathbf{c} = \left[C_1\ C_2\ ...\ C_{N_{char}}\right].$$

The Sync Character C_{sync} is then appended at the beginning of the text message to form the block:

$$\mathbf{d} = \left[C_{sync}C_1\ C_2\ ...\ \ C_{N_{char}}\right], \quad |\mathbf{d}| = N_{char} + 1 = N_{block},$$

so that $d_i \in \mathbb{V} - \{C_{sync}\}$, where d_i is the i-th element of $\mathbf{d}$. The block's maximum size is fixed and known to the receiver, in order to perform synchronization and Repetition Decoding at the same time.
In order to transmit a message over a noisy channel that may corrupt the transmission in a few places, repetition coding is utilized to enhance its performance. The block $\mathbf{d}$ is repeated N_{rep} times and then concatenated, thus creating the character buffer **TxB** of size $N_{buf} = N_{rep} \cdot N_{block}$, which includes the total packets to be transmitted.

Packet Division and ASCII Encoding

Since letters are the structural components of text, each character forms a packet to be transmitted. Every character of the text buffer is encoded in 7-bit ASCII code to obtain the binary vectors $\mathbf{b}$ of each packet:

$$\mathbf{b}[n] = \{0,1\}^{7\times 1}, \quad n = 1, 2, ..., N_{buf}.$$

Symbol Mapping

The entries of each vector **b** are mapped to symbols, defining the vector:

$$s_{i,1} = \begin{cases} -1, & \text{if } b_{i,1} = 0 \\ +1, & \text{if } b_{i,1} = 1 \end{cases}, \quad i = 1, 2, ..., N_{sym},$$

where $N_{sym} = 7$. The Symbol Rate R_{sym} is fixed at 441 symbols per second, for reasons that will be discussed below, and Symbol Period $T_{sym} = \frac{1}{R_{sym}}$.

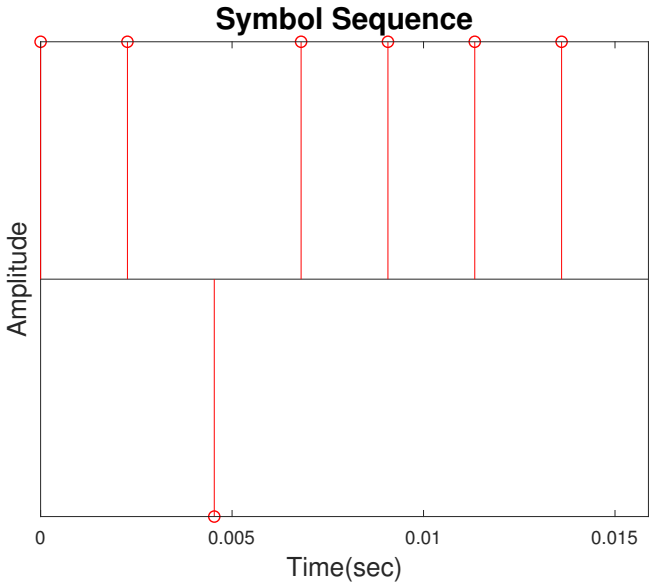

Figure 2.2: Symbol Sequence of Sync Character

Spreading Sequence Generation using Linear Feedback Shift Register

The LFSR generates M distinct binary sequences $\mathbf{q}_{N_{Seq} \times 1}$ where M is the period of the LFSR for the corresponding register length N_{Seq}, s.t : $M \leq 2^{N_{Seq}} - 1$ [6], satisfying the equality iff the feedback polynomial is primitive. Interestingly enough, there may be more than one primitive polynomial resulting in a maximal LFSR period, for a given register's length. The secret key, known

to both the transmitter and the receiver, contains both the taps and the positions of the high bits in the initial state. In order to achieve a reliable LFSR period for this work, we set $N_{Seq} = 20$. The following sequence matrix is generated:

$$\mathbf{Q} \in \{0,1\}^{N_{Seq} \times M}.$$

The entries of this matrix are mapped to symbols to obtain:

$$R_{i,\,j} = \begin{cases} +1, & \text{if } Q_{i,j} = 0 \\ -1, & \text{if } Q_{i,j} = 1 \end{cases}, \quad i = 1,\ 2,\ \dots\ ,\ M\ , \quad j = 1,\ 2\ , \dots\ ,\ N_{Seq}.$$

The M fixed-size sequences $\mathbf{r}$ are extracted from matrix $\mathbf{R}$, being its M column vectors: $\mathbf{r}_{N_{Seq} \times 1}[k]$, where $k = 1, 2, \dots, M$.

Direct Sequence Spread Spectrum (DSSS)

DSSS [7] is the technique which gave birth to widely known Code Division Multiple Access (CDMA). It is a Spread Spectrum modulation technique used to reduce overall signal interference [8] .
The Kronecker Tensor Product of the symbol vector and spreading sequence is computed to obtain the spread symbol vector:

$$\mathbf{b}_{SS} = \mathbf{s}_{7\times 1} \otimes \mathbf{r}_{N_{seq} \times 1}[k], \qquad k = 1,\ 2,\ \dots\ ,\ M,$$

where $|\mathbf{b}_{SS}| = N_{sym} \cdot N_{Seq} = N_{chip} = 140$.
DSSS offers a processing gain $PG = 10 \log_{10}(N_{seq})$ at the receiver side as, during the inverse procedure, interference power is spread across the band by the spreading code, while the desired signal is restored back to its original spectrum.

Chip Rate is defined as $R_{chip} = R_{sym} \cdot N_{Seq} = 8820$ chips per second and Chip Period as $T_{chip} = \frac{1}{R_{chip}}$ seconds.

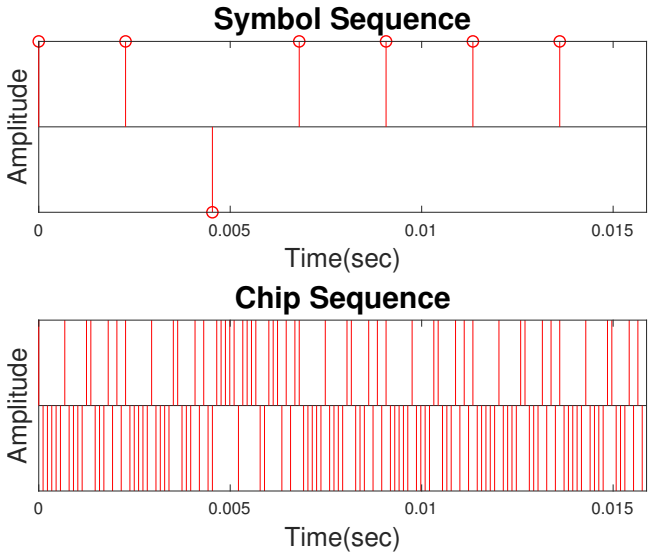

Figure 2.3: Symbol vs Chip Sequence

Pulse Shaping

In order to match audio sampling frequency, we set $\mathrm{F_s} = 44100$ kHz and sampling period $\mathrm{T_s} = \frac{1}{\mathrm{F_s}}$. As there is a restriction in Audio Bandwidth, the signal to be transmitted must have a BW $\leq$ 20kHz.
In order to perform matched filtering, a Square-Root Raised Cosine (SRRC) filter is used. The impulse response of such a filter can be given as [9]:

$$\mathrm{h}(\tau) = \frac{4a}{\pi\sqrt{\mathrm{T}}} \frac{\frac{1}{4a\frac{\tau}{\mathrm{T}}}\sin\left[\pi\frac{\tau}{\mathrm{T}}(1-a)\right] + \cos\left[\pi\frac{\tau}{\mathrm{T}}(1+a)\right]}{1-\left(4a\frac{\tau}{\mathrm{T}}\right)^2},$$

where $\alpha \in [0,1]$ is the roll-off factor and new symbol (chip) period $\mathrm{T} = \mathrm{T_{chip}}$. The function $\mathrm{h}(\tau)$ is called a Square-Root Raised Cosine and is extremely important for digital telecommunications [9].
A great portion of the audio spectrum has to be used, in order for the message signal to attain a noise-like form after the pulse shaping process. For roll-off factor a = 1, $\mathrm{H_{SRRC}(f)} = \mathcal{F}\left\{\mathrm{h}(\tau)\right\}$ becomes a pure raised-cosine shaped frequency response with no "flat-top" and with the widest bandwidth of the family: $\mathrm{BW_{H_{SRRC}}} = \frac{1}{\mathrm{T_{chip}}}$.

To simulate the continuous time convolution, we define the truncated SRRC pulse, shown in Fig 2.4, with a duration of 2AT, where A = 4 is half the duration of the pulse in chip periods and T is the chip period:

$$\widehat{\mathrm{h}}(\tau) = \begin{cases} \mathrm{h}(\tau), & \text{if } \tau \in [-\mathrm{AT}, +\mathrm{AT}] \\ 0, & \text{otherwise} \end{cases}$$

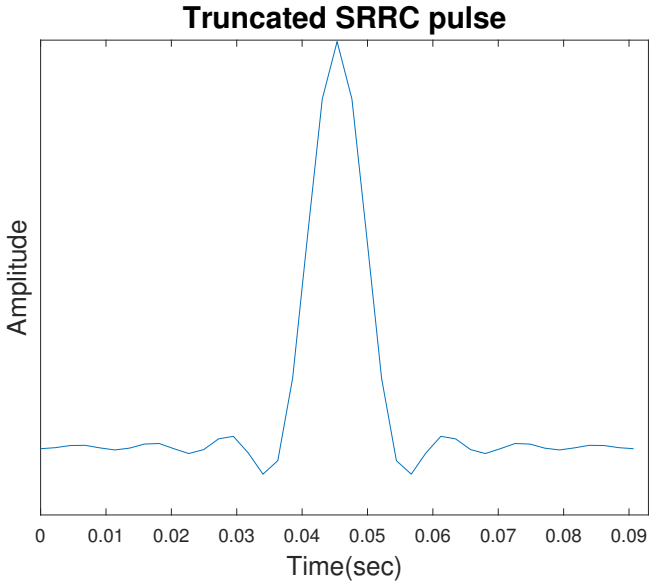

Figure 2.4: Truncated SRRC pulse

To simulate the continuous time signal $\mathrm{X}_\delta(\mathrm{t}) = \sum_{\mathrm{k}=0}^{\mathrm{N}-1} \mathbf{b}_{\mathrm{ss}}\delta(\mathrm{t}-\mathrm{kT})$ before pulse shaping, the chip vector has to be upsampled. In order to keep the resulting filtered audio signal's bandwidth below 10 kHz for reasons that will be discussed below, an extremely short vector $\mathbf{v}$ is selected as the oversampling factor of size $\mathrm{N}_{\mathrm{over}} = \frac{\mathrm{T}_{\mathrm{chip}}}{\mathrm{T}_{\mathrm{s}}} = \frac{\mathrm{F}_{\mathrm{s}}}{\mathrm{R}_{\mathrm{chip}}} = \frac{44100}{8820} = 5$, as the signal has to be short in time duration but great in bandwidth. Let :

$$\mathbf{v} = [1\,0\,0\,0\,0\,]^{\intercal}, \quad |\mathbf{v}| = \mathrm{N}_{\mathrm{over}} = 5.$$

The upsampled chip vector is given by the tensor product of the chip vector

and the oversampling vector:

$$\mathbf{b}_{\text{ups}} = \mathbf{b}_{\text{SS}} \otimes \mathbf{v}, \quad |\mathbf{b}_{ups}| = \text{N}_{\text{chip}} \cdot \text{N}_{\text{over}} = \text{N}_{\text{ups}} = 700.$$

Finally, the upsampled chip vector is convoluted with the truncated SRRC vector $\mathbf{h}$, to form the audio message signal in Fig. 2.5:

$$\mathbf{x}_{\text{aud}} = \mathbf{b}_{\text{ups}} * \mathbf{h} \ ,$$

where $|\mathbf{x}_{\text{aud}}| = \text{N}_{\text{over}}\big(\text{N}_{\text{chip}} + 2\text{A}\big) = 740 = \text{N}_{\text{aud}}$ (because of the delay caused by the convolution), with a bandwidth $\text{BW}_{\mathbf{x}_{\text{aud}}} = \frac{1}{\text{T}_{\text{chip}}} = 8820\text{Hz}$.

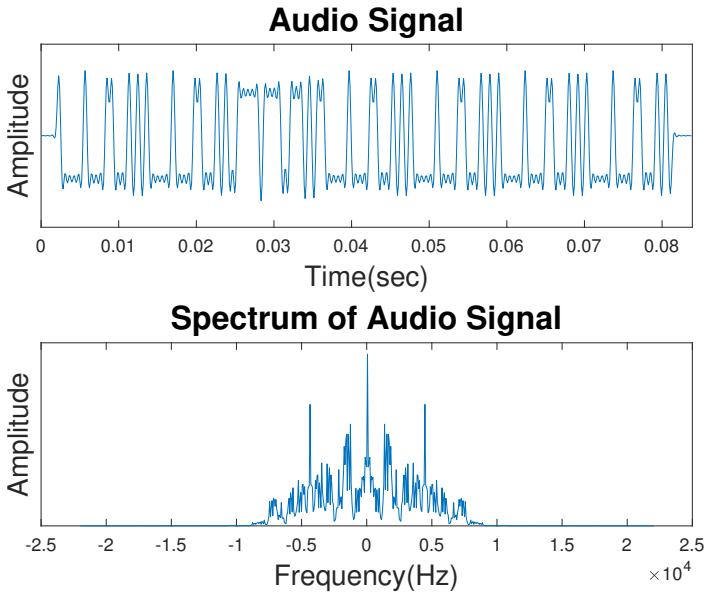

Figure 2.5: Audio Signal

Upconversion

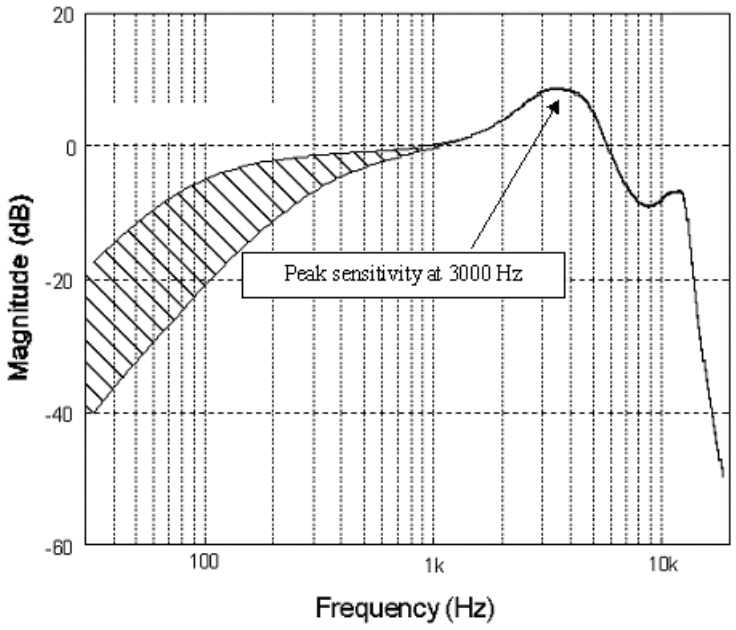

Figure 2.6: Human Hearing Sensitivity vs Frequency [2]

Because the Human Auditory System is less sensitive to higher frequencies near its limit of 20 kHz as shown in Fig. 2.6, in conjunction with the fact that audio bass is often pre and post-amplified, it would be beneficial to raise $\mathbf{x}_{\text{aud}}$ in the audio frequency domain by mixing it with a carrier $\cos(2\pi \mathrm{F_{up}} t)$. When these signals are mixed [10], two additional signals are formed, one at $+\mathrm{F_{up}}$ and one at $-\mathrm{F_{up}}$. In order to center the upconverted signal at a higher audio frequency, to which the Human Auditory System will be less sensitive, as well as to avoid aliasing caused by the idol at $-\mathrm{F_{up}}$, it is required that :

$$\mathrm{F_{up}} > \mathrm{BW}_{\mathbf{x}_{\text{aud}}}.$$

Because of the above, the upconverted audio signal will have a total bandwidth $\mathrm{BW}_{\mathbf{x}_{\text{up}}} = 2\ \mathrm{BW}_{\mathbf{x}_{\text{aud}}} = 17640$ Hz. That is also why we insisted on limiting $\mathrm{BW_{aud}} < 10\text{kHz}$ to avoid aliasing. The value of $\mathrm{F_{up}}$ has to ensure that the audio message signal is not cut by the low pass filter at the receiver side, but at the same time we have to make the most of the available

bandwidth. The upconverting carrier frequency is given by :

$$\mathrm{F_{up}} = 20000 - \mathrm{BW_{\mathbf{x}_{aud}}} = 11180 \text{ Hz}.$$

Finally, the upconverted audio *message* signal, shown in Fig. 2.7, is given by:

$$\mathbf{x}_{\mathrm{up}} = \mathbf{x}_{\mathrm{aud}}(\mathrm{nT_s}) \cdot \cos\left(2\pi \mathrm{F_{up}} \mathrm{nT_s}\right), \quad \mathrm{n} = 1, 2, ..., \mathrm{N_{aud}}.$$

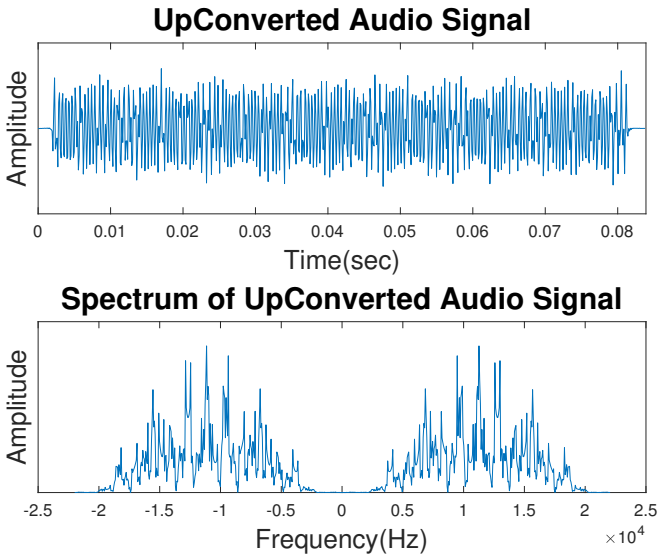

Figure 2.7: Upconverted Audio Signal

Host Audio Signal's Power Approximation

As the *Host* audio signal is live radio broadcast, neither its content nor its power can be known a priori. A statistical estimation of its average power $\mathrm{P_{est}}$ has been performed on a variety of prerecorded radio broadcast audio signals. This includes both talk shows and different kinds of music. Furthermore, it is found that a great portion of the host's average power is concentrated below 4kHz, thus justifying the Upconversion step, as interference caused by the *Host* audio signal will now be avoided.

Average Power Normalization

The last step, before embedding process, is average power normalization. It is crystal clear that message signal's instantaneous amplitude must be way lower than the host signal's in order to be inaudible by a third party.
The average power of the upconverted *message* audio signal $\mathbf{x}_{\mathrm{up}}$ is given by:

$$\mathrm{P_{up}} = \frac{1}{\mathrm{N_{aud}}} \sum_{\mathrm{n=1}}^{\mathrm{N_{aud}}} \mathbf{x}_{\mathrm{up}}^2(\mathrm{nT_s}).$$

The normalized *message* audio signal, so that $\mathrm{P_{norm}} = \mathrm{P_{est}}$, is calculated by:

$$\mathbf{x}_{\mathrm{norm}} = \sqrt{\frac{\mathrm{P_{est}}}{\mathrm{P_{up}}}}\ \mathbf{x}_{\mathrm{up}}.$$

We define the received $\mathrm{SNR_{notag}} = \frac{\mathrm{P_{host}}}{\mathrm{P_{noise}}}$, which indicates how strong the radio broadcast *Host* audio signal is in relation to audio *noise*, that is when the tag is not operating. $\mathrm{P_{host}}$ and $\mathrm{P_{noise}}$ refer to the Host audio signal's and audio noise power, respectively. We also define the Concealment Loss in power: $\lambda = \frac{\mathrm{P_{norm}}}{\mathrm{P_{host}}}$, describing how weak the normalized *message* audio signal is compared to the *Host* audio signal. We have to be careful when selecting a value for λ to scale $\mathbf{x}_{\mathrm{norm}}$, as there is a trade-off between Error Rate and Inaudibility. The final audio message signal is given by :

$$\mathbf{x}_{\mathrm{msg}} = \sqrt{\lambda}\ \mathbf{x}_{\mathrm{norm}}.$$

Now, the audio signal $\mathbf{x}_{\mathrm{msg}}$ is ready to be embedded to the *Host* audio signal.

2.2 Embedding Process

Given the $\mathrm{SNR}_{\mathrm{notag}}$ defined in the previous section , we obtain $\mathrm{P}_{\mathrm{noise}}$ and generate the zero-mean Additive White Gaussian Noise vector:

$$\mathbf{n} \sim \mathcal{N}(0, \mathrm{P}_{\mathrm{noise}}) \ , \quad \mathrm{n} = 1,\ 2,\ \ldots\ , \mathrm{N}_{\mathrm{aud}}.$$

As shown in 1.3 , the RF signal corruption by noise is resulting in audio noise being added to the Host audio signal at the receiver's output, forming the audio station signal :

$$\mathbf{x}_{\mathrm{fm}} = \mathbf{x}_{\mathrm{host}}(\mathrm{nT_s}) + \mathbf{n}(\mathrm{nT_s}) \ , \quad \mathrm{n} = 1,\ 2,\ \ldots\ , \mathrm{N}_{\mathrm{aud}}.$$

Finally, the embedding process is reduced to a Baseband system model and concluded by the addition of the two audio signals, as described in section 1.3 and [1] , forming the *stego* audio signal show in Fig. 2.8:

$$\mathbf{y}_{\mathrm{stego}} = \mathbf{x}_{\mathrm{msg}} + \mathbf{x}_{\mathrm{fm}}.$$

The received audio signal now undergoes the inverse process in baseband to retrieve the concealed text message.

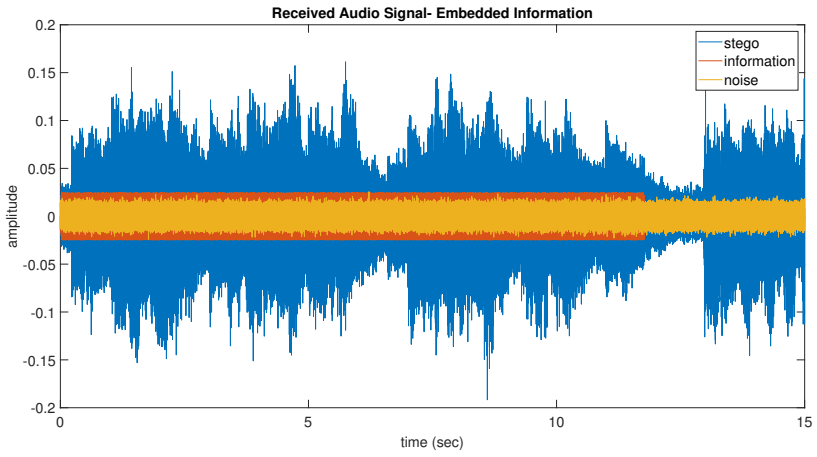

Figure 2.8: Stego Audio in time

2.3 Receiver Design

Synchronization

Although first packet detection could be achieved through energy peak detection when the tag starts operating, alterations to the symbol period were observed in the actual implementation. That is the reason a Sync Word has been used to synchronize the receiver after every block reception.
The Sync Word is placed at the beginning of a block. A moving data window, corresponding to the respective Sync Word generated by the LFSR at the receiver side and equal to the size of the packet in samples, is shifted over the received data as their correlation is being computed.
At first, a Sync Word has to be detected. When the correlation coefficient is above a predefined threshold, the receiver announces the detection of a packet. Since the block's maximum size is fixed and known to the receiver, the number of packets after which the next Sync Word is expected to be placed is also known a priori. Then, the same process is repeated, setting the frame most correlated with the Sync Word as the beginning of each block.

Downconversion

Having discarded the Sync Word and as the packet length $\mathrm{N_{aud}}$ is known, the following received character packet $\mathbf{y}_{\mathrm{pack}}$ is mixed with a carrier of the same frequency F_{up}. The signal previously centered at $+\mathrm{F}_{up}$ is forming one signal at $2\mathrm{F}_{up}$ and one at 0, while the signal previously centered at $-\mathrm{F_{up}}$ is forming one idol at 0 and another one at $-2\mathrm{F_{up}}$. We get the downconverted audio signal by the following equation:

$$\mathbf{y}_{\mathrm{down}} = \mathbf{y}_{\mathrm{pack}}(\mathrm{nT_s}) \cdot \cos\left(2\pi \mathrm{F_{up}} \mathrm{nT_s}\right), \quad \mathrm{n} = 1,\ 2,\ \ldots\ , \mathrm{N_{aud}}.$$

Matched Filtering

Since the SRRC pulse is symmetric, $\mathrm{h}(\tau) = \mathrm{h}(-\tau)$, the same SRRC pulse shaping function will be used for the matched filter. Because the SRRC filter is low pass, the signal at $2\mathrm{F}_{up}$ is rejected and $-2\mathrm{F}_{up}$ does not practically exist. The downconverted audio signal is convolved with the truncated SRRC filter's discrete time impulse response $\overleftarrow{\mathbf{h}}$ and the output of the matched filter is given by:

$$\mathbf{z} = \mathrm{T_s} \left[\mathbf{y}_{\mathrm{down}} * \overleftarrow{\mathbf{h}} \right],$$

which is a tight approximation of continuous time convolution.

Sampling

Because of two consecutive convolutions, the filtered vector is now delayed by a whole duration of the truncated SRRC pulse in chip periods:

$$\mathbf{r}_{\mathrm{x}}[\mathrm{n}] = \mathbf{z} \left[\mathrm{N_o}\big(2\mathrm{A} + (\mathrm{n}-1)\big) + 1 \right], \quad \mathrm{n} = 1,\ 2,\ \ldots\ , \mathrm{N_{chip}}.$$

Inverse Direct Sequence Spread Spectrum (IDSSS)

The following maximum likelihood decision rule is defined:

$$\mathrm{DR}(\mathrm{x_j}) = \begin{cases} -1, & \text{if } \mathrm{x_j} \leq 0 \\ +1, & \text{if } \mathrm{x_j} > 0 \end{cases}$$

The decision rule is applied on the received samples and the resulting chip vector is reshaped into a matrix $\mathbf{U} \in \{\ \mathrm{DR}(\mathbf{r}_{\mathrm{x}})\ \}^{7\times \mathrm{N_{seq}}}$.
Given the secret key, the receiver can generate the same binary sequences because the process is deterministic, and map them to symbols, thus creating the matched spreading sequences.
Then, $\mathbf{U}$ is multiplied with its respective symbol-mapped spreading sequence

generated by the LFSR, to obtain the symbol vector:

$$\mathbf{s}_{7\times1} = \mathbf{U} \times \mathbf{r}[\mathrm{k}], \quad \mathrm{k} = 1, 2, ..., \mathrm{M}$$

Symbol to Bit Conversion

At this point, the entries of each symbol vector **s** are mapped back to bits, defining the vector:

$$\mathrm{b}_{\mathrm{i},1} = \begin{cases} 1, & \text{if } \mathrm{s}_{\mathrm{i},1} = +1 \ , \\ 0, & \text{if } \mathrm{s}_{\mathrm{i},1} = -1 \ , \end{cases} \quad \mathrm{i} = 1, 2, ..., \mathrm{N}_{\mathrm{sym}}.$$

ASCII Decoding

Again, 7-bit ASCII code is used to decode the bit vector **b** into a character $\mathrm{C} \in \mathbb{V}$, filling the buffer:

$$\mathbf{RxB} = \left[\mathrm{C}_1 \ ... \ \mathrm{C}_\mathrm{L} \ \mathrm{C}_{\mathrm{L}+1} \ ... \ \mathrm{C}_{2\mathrm{L}} \ ... \ \mathrm{C}_{(\mathrm{R}\text{-}1)\mathrm{L}+1} \ ... \ \mathrm{C}_{\mathrm{RL}}\right],$$

where $\mathrm{L} = \mathrm{N}_{\mathrm{block}}$ and $\mathrm{R} = \mathrm{N}_{\mathrm{rep}}$.
The Sync Character is placed in positions $\kappa \ \mathrm{L} + 1$, where $\kappa = 0, 1, ..., \mathrm{R} - 1$.
The received Character Buffer is formed by removing the Sync Characters:

$$\mathbf{CharB} = \left[\mathrm{C}_2 \ ... \ \mathrm{C}_\mathrm{L} \ \mathrm{C}_{\mathrm{L}+2} \ ... \ \mathrm{C}_{2\mathrm{L}} \ ... \ \mathrm{C}_{(\mathrm{R}\text{-}1)\mathrm{L}+2} \ ... \ \mathrm{C}_{\mathrm{RL}}\right],$$

which is redefined as :

$$\mathbf{CharB} = \left[\mathrm{C}_1 \ ... \ \mathrm{C}_\mathrm{K} \ \mathrm{C}_{\mathrm{K}+1} \ ... \ \mathrm{C}_{2\mathrm{K}} \ ... \ \mathrm{C}_{(\mathrm{R}\text{-}1)\mathrm{K}+1} \ ... \ \mathrm{C}_{\mathrm{RK}}\right],$$

where $\mathrm{K} = \mathrm{N}_{\mathrm{char}}$, which is the length of the secret text that was transmitted. At this point, the Character Buffer contains the secret text message repeated R times.

Repetition Decoding

The *distinct* characters that appear in the respective positions of each block are referred to as $\widehat{\mathrm{C}}$. Performing majority voting, the frequency of *distinct* characters $\widehat{\mathrm{C}} \in \widehat{\mathbb{V}} \subseteq \mathbb{V}$ is computed and the most frequent character is selected, to create the text buffer $\mathbf{T}$, admitting to the following formula:

$$\mathbf{T}_{\mathrm{i}} = \underset{j}{\operatorname{argmax}}\ \mathrm{freq}(\widehat{\mathrm{C}}_{\mathrm{j}}),$$

where $\mathrm{i} = 1, 2, ..., \mathrm{K}$ and $\mathrm{j} = \mathrm{i},\ \mathrm{K{+}i},\ \ldots,\ (\mathrm{R\text{-}1})\mathrm{K} + \mathrm{i}$.
Finally, the text buffer $\mathbf{T}$ consisting of the decoded *text* message is defined as:

$$\mathbf{T} = \left[\ \widehat{\mathrm{C}}_1\ \widehat{\mathrm{C}}_2\ ...\ \widehat{\mathrm{C}}_{\mathrm{N_{char}}}\right].$$

Chapter 3

Implementation

3.1 Transmitter

Digital Audio File Generation

The vector of audio data $\mathbf{x}_{\text{msg}}$ is written to a file using Matlab's built-in *audiowrite* function with a sample rate $F_s = 44100$Hz. A single audio channel is used (mono) and the bit depth is set at 16 bits per sample. Instead of writing the audio data to an MP3 file, which is the most popular digital audio format, the .wav format is selected, as shown in Fig. 3.2. An MP3 file is compressed and lossy whereas a WAV file is lossless and uncompressed. A 44100Hz 16-bit WAV has a full frequency response up to 22KHz where as an MP3 cuts off around the 18KHz mark. That would be careless for our purpose because the band 18-20 kHz would be rejected despite being occupied, as the maximum frequency of the upconverted signal is 20kHz.

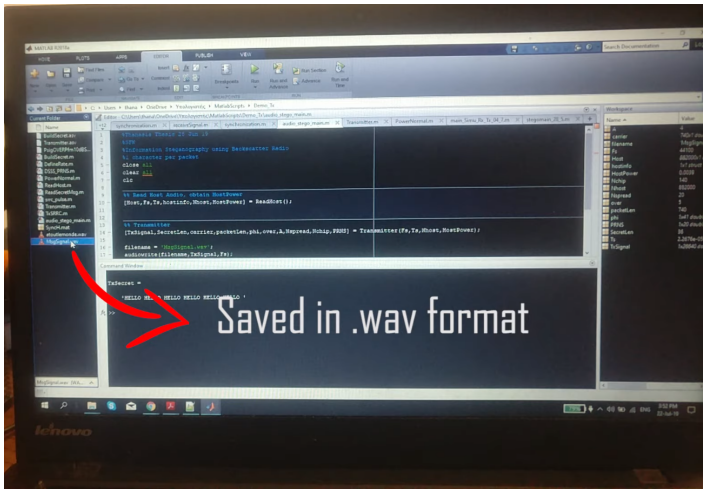

Figure 3.1: Audio Samples saved in .wav format

Signal Amplifying

A laptop is used as a playback device, shown in Fig. 3.2. Its sound card uses a digital-to-analog converter (DAC), which converts the generated digital signal data into an analog signal.

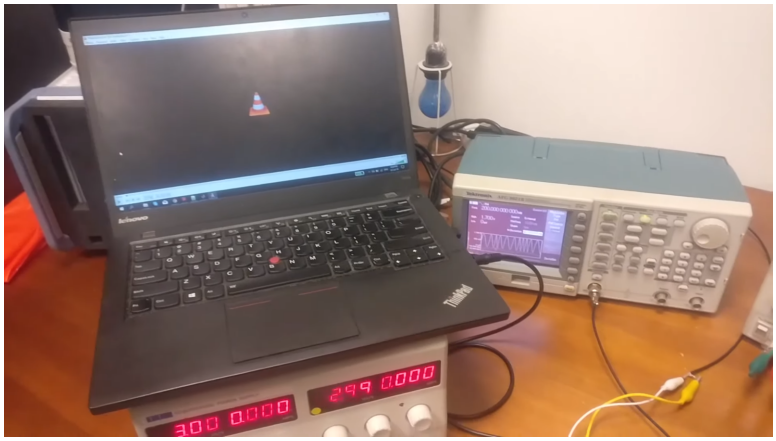

Figure 3.2: Playback Device / Digital to Analog Converter

The output analog signal is connected to an operational amplifier circuit via a male 3.5 mm plug to plug audio jack. The OpAmp provides a gain $G = \frac{R_f}{R_i}$, and the laptop's output volume gain is fixed at such a level, so that the analog signal fed to the function generator has a maximum amplitude of 1 Volt to set a frequency deviation of 42 kHz, as explained in section 1.3.

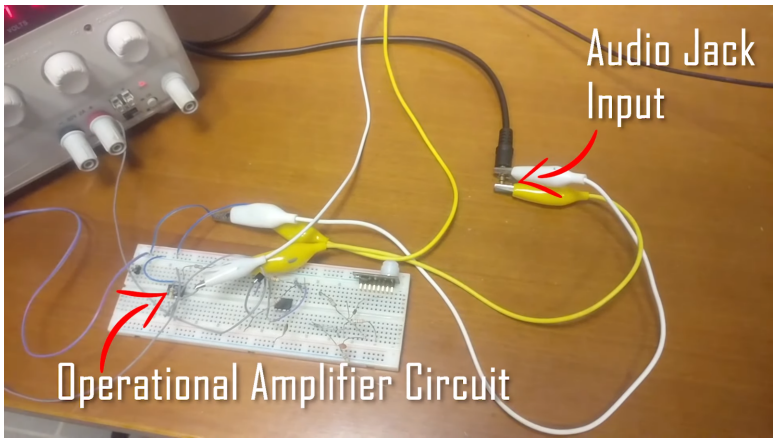

Figure 3.3: Operational Amplifier Circuit

FM Backscatter Tag

The amplified analog signal controls a function generator and gets FM modulated on a F_{sw} = 200 kHz sine wave, for reasons that will be discussed in 3.2, with a frequency deviation of 42 kHz, , as shown in Fig. 3.4.

Figure 3.4: Function Generator as FM modulator

The FM modulated audio signal is driving an RF switch, shown in Fig. 3.5, terminating an antenna to different loads, thus forcing it to backscatter the FM message signal. During operation, an illuminating FM station signal with a carrier frequency $F_{station}$ impinges on the antenna, resulting in a signal being backscattered. As explained in 1.3, the backscattered signal is the FM *stego* signal, centered both at $F_{station} + F_{sw}$ and at $F_{station} - F_{sw}$.

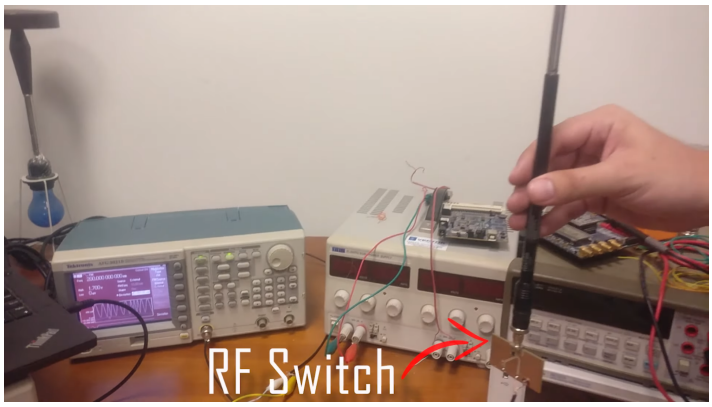

Figure 3.5: RF Switch [1]

Tx Software Overview

The implementation software of the Transmitter is presented in Fig. 3.6. The user is asked to input a text message and a private key. The output is a vector containing the produced audio samples, which is then written to a WAV audio file.

```
%Information Steganography using Backscatter Radio / Baseband System Model
%Transmitter

%% Insert secret text message
TxMsg='PLANT THE BOMB AT 221B BAKER STREET. CALL MORIARTY. ANARCHY IN THE UK!';

%% Define SNR and BitRate
SNRnotag_db=10;
BitRate=441;

%% Statistical Estimation of Avg HostPower
[AvgHostPower]=EstHostPower();

%% Import Host Audio Signal
[Host,Fs,hostinfo,Nhost] = ReadHost(); %Convert from Stereo to Mono

%% Construct TxBuffer
TxBlock=[SyncWord TxMsg];
RepeatNum=10;
TxBuff=repmat(TxBlock,[1 RepeatNum]); %Repetition Encoding
Sig=zeros(Nhost,1);%right zero-padded Msg Signal

%Create LFSR Object, Determine taps and initial state using a secret key
[LFSR]=CreateLFSR(key);

%% Construct Upconverted Audio Message Signal
for i=1:length(TxBuff)

    character=TxBuff(i);

    [Xup,LFSR]= Convert2Audio(character,Fs,BitRate,LFSR); % Form each packet

    Sig( length(Xup)*(i-1)+1: length(Xup)*i )= Xup ; % Concatenate packets
end

%% Add AWGN to audio Host signal
SNRnotag=10^(SNRnotag_db/10);
noise=sqrt(AvgHostPower/SNRnotag)*randn(size(Host));
Xfm=Host+noise;

%% Average Power Normalization, Set Pmsg < Pnoise
[Xmsg,Pmsg]= PowerNormal(Sig,AvgHostPower,SNRnotag);

%% Embedding Process
stego=Xmsg + Xfm;
```

Figure 3.6: TxSoftware Overview

3.2 Receiver

Receiver Front End

A handheld smart phone equipped with FM radio is used as an FM receiver to demodulate the FM backscattered signal and record the resulting *stego* audio signal in real time. The local FM radio station "ERA SPORT" with a carrier frequency of 90.1 MHz has been convenient for the purpose of experiments by providing both an excellent reception and a satisfying frequency range on each side, where no interference was caused by another station, as the closest carriers were at 89.6 MHz and 91.0 MHz, respectively. The receiver is tuned at $F_{station} - F_{msg} = 90.1 - 0.2 = 89.9$ MHz. The distance between the tag and receiver antenna during the conducted experiments was in the order of 10 centimeters.

Figure 3.7: FM Receiver / Recorder

Receiver Audio Signal Processing

The *stego* audio signal is recorded by the smart phone and saved in .wav format, presented in Fig. 3.8. Then, it is imported to Matlab with the use of built-in *audioread* function, which outputs a vector containing the audio

samples of *stego*, sampled by a rate F_s :

$$[\ \text{stego}, F_s\ ] = audioread\ \ (\ \text{'stego.wav'}\ )$$

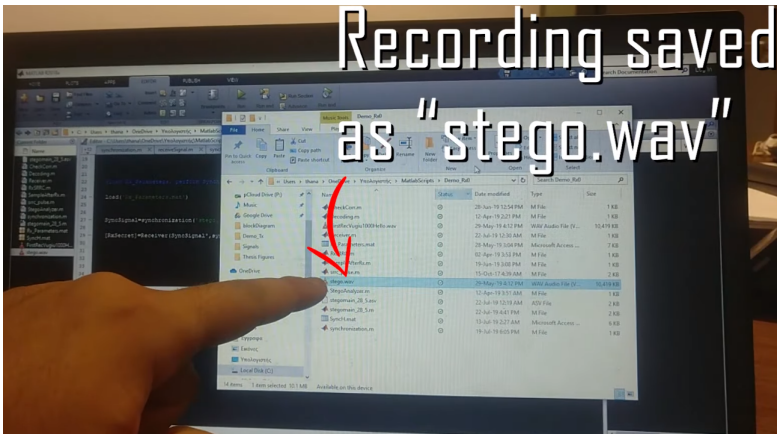

Figure 3.8: Recording saved in .wav format

Now, stego can be processed in baseband as described in section 2.3 to retrieve the transmitted text message. The correct private key containing both the LFSR seed and taps defining each output sequence will create a matched LFSR, thus achieving reliable and secure communication. The taps is a vector containing the polynomial coefficients equal to 1 and the seed is a binary vector determining the register initial state, as shown in in Fig. 3.9.

```
key =

  struct with fields:

    taps: [20 17]
    seed: [1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 0 1 0 1]

>> Nseq

Nseq =

    20
```

Figure 3.9: Private Key Format

RxSoftware Overview

The implementation software of the Receiver is given below. The user is asked to input the secret key, whereas the vital parameters defining the receiver configuration are built-in. Finally, the output is the decoded text message.

```
%Information Steganography using Backscatter Radio / Baseband System Model
%Receiver

%% Load Receiver Parameters
load('Rx_Parameters.mat');

%% Create Rx LFSR Object using the secret key
[LFSR]= CreateLFSR(key,RxParam);

%% Detect First Packet (First Sync Word)
[SyncPos,LFSR]= JumpStart(stego,RxParam,LFSR);

PackPresenceFlag=1; %Turns to zero when reception is completed

BlockCnt=0;
while(PackPresenceFlag)

    BlockCnt=BlockCnt+1;

%% Character Block Construction
    [CharBlock(BlockCnt),LFSR]= ConvertAudio2Text(stego,RxParam,LFSR,SyncPos);


%% Synchronization
    [SyncPos,LFSR]=Synchronization(stego,RxParam,LFSR,SyncPos);
    if SyncPos<0 %If Correlation is below a threshold
        PackPresenceFlag=0; % Break
    end
end

%% Repetition Decoding
[TextMsg]= RepDec(CharBlock);
```

Figure 3.10: RxSoftware Overview

Chapter 4

Results

In the current section, the performance of our stegosystem is studied through numerical results in a simulation environment. It has to be noted that the Processing Gain offered by the DSSS has not been taken into account during this performance analysis.
The number of packets constituting the secret message is 70, to which a (10,1) repetition code is applied, resulting in 700 total packets being transmitted. The number of Monte Carlo experiments performed is 7000, an order of magnitude larger than the number of transmitted packets. We define $\mathrm{SNR}_{\mathrm{notag}}$ (introduced in 2.1) as the received SNR when the tag is not operating, meaning it is just the noisy host signal, referred to as *received station* signal.

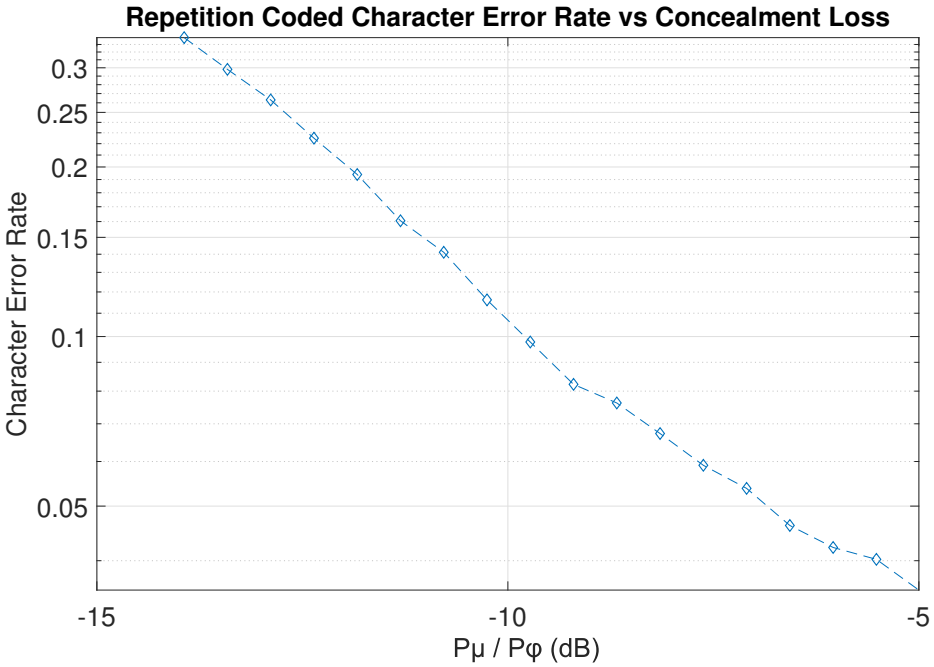

Figure 4.1: Character Error Rate vs Concealment Loss

Fig. 4.1 plots the Character Error Rate of the stegosystem, having utilized the repetition coding, against the Concealment Loss, defined in 2.1, for different values of the message audio signal's power. Making use of the conclusion reached in section 1.3, the Concealment Loss expresses the ratio of message/information audio signal's μ(t) power P_μ over the power P_ϕ of the host audio signal ϕ(t). The results are reasonable for our AWGN communication model as a rapid decrease of the Character Error Rate occurs as the message audio signal's power increases. The break of the curve shows that the probability of error drops down to 10% for Concealment Loss $\lambda = -10$ dB, when $\text{SNR}_{\text{notag}} = 15$ dB. That probability of Character Error is justified as a character is consisting of 7 symbols, meaning that even if one of them is received falsely, then the character is translated with incorrectly. While the error rate varies to satisfactory levels, we have to define an Imperceptibility metric to ensure that the information audio signal remains relatively inaudible.

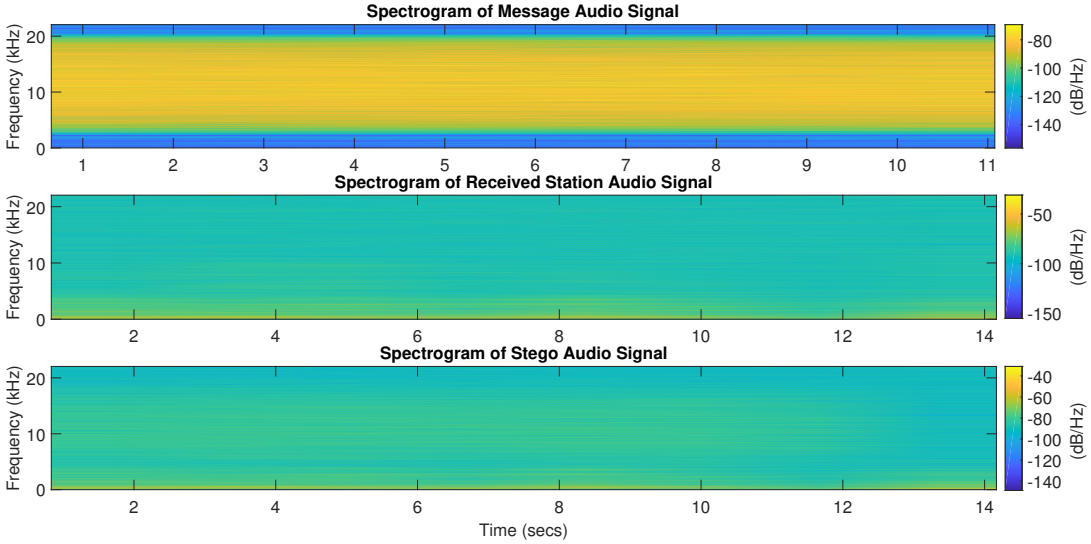

Figure 4.2: Spectrograms of Audio Signals

Fig. 4.2 presents the spectrograms of the *message*, *station* and *stego* audio signals, respectively, from top to bottom, for a Concealment Loss $\lambda = -10$ dB and $\text{SNR}_{\text{notag}} = 15$ dB. In the first spectrogram, the message audio signal is truncated after the 11 second mark in order to examine the stego's response

comparatively, when the tag is off or on. In case of ensured imperceptibility, we expect the received station audio signal's spectrogram (that is, when the tag is off) to be similar to stego's spectrogram (when the tag is on). Apparently, the results are good, as the provided visual representation indicates that an effective concealment has been performed. It is easy to deduce that the embedded information has no effect whatsoever to the host signal, making its presence undetectable. Remember that the aforementioned power conditions result in a Character Error Rate of 10% , as shown in Fig. 4.1.

Chapter 5

Conclusions and Future Work

Conclusion

In this work, information converted to an audio signal is successfully conveyed covertly using FM ambient backscatter radio. In the extreme case that its presence is detected by a third party using a spectrum analyzer, the information is protected by a private-key encryption layer. That is provided by a Linear Feedback Shift Register generating spreading sequences to implement the Direct Sequence Spread Spectrum technique of Audio Steganography. If the receiver knows the key, he can create a matched LFSR to retrieve the information. Given the low rate requirements, repetition coding was exploited to enhance the performance.

Future Work

As future work , more sophisticated error correction codes shall be implemented and additional pulse shaping functions have to be examined. Long PN sequences have to be used as spreading sequences because of their excellent mathematical properties. Furthermore, FM station signals could be streamed to the transmitter in real time, for the demodulated station audio signal to be taken into account and enhance the step of power normalization, as getting more recent information about the host audio signal's average power would result in a more accurate concealment of the information signal compared to a statistical approach. Finally, a relatively small-sized application specific integrated circuit (ASIC) with reduced power supply could be designed, in order to get rid of the massive devices that were used in our implementation, thus minimizing the risk of arising suspicion during communication.

Bibliography

[1] G. Vougioukas and A. Bletsas, "24 μ watt 26m range batteryless backscatter sensors with fm remodulation and selection diversity," in *2017 IEEE International Conference on RFID Technology & Application (RFID-TA)*, Warsaw,Poland, Sep. 2017, pp. 237–242.

[2] Antonine-Education. (2016) Ear sensitivity vs frequency on a fixed reference level. [Online]. Available: http://www.antonine-education.com/Image_library/Physics_5_Options/Medical_Physics/f-response.gif

[3] R. Tanwar and M. Bisla, "Audio steganography," in *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, Feb 2014, pp. 322–325.

[4] A. J. Viterbi, *Principles of Spread Spectrum Communication.* Addison-Wesley Publishing Company, 1995.

[5] S. W. Golomb, "Shift-register sequences and spread-spectrum communications," in *Proceedings of IEEE 3rd International Symposium on Spread Spectrum Techniques and Applications (ISSSTA'94)*, July 1994, pp. 14–15 vol.1.

[6] J. Bai, T. Zhang, X. Yu, and Y. Wang, "The estimation of the pn sequence's period of the dsss signals in narrowband interference environment," in *2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Sep. 2011, pp. 1–4.

[7] F. Djebbar, B. Ayad, H. Hamam, and K. Abed-Meraim, "A view on latest audio steganography techniques," in *2011 International Conference on Innovations in Information Technology*, April 2011, pp. 409–414.

[8] N. Cvejic, "Algorithms for audio watermarking and steganography," Ph.D. dissertation, University of Oulu, Oulu, Finland, 2004.

[9] A. Lapidoth, *A Foundation in Digital Communication*. Cambridge University Press, 2017.

[10] B. Razavi, *RF Microelectronics*. New Jersey: Prentice-Hall, 1998.