Near-ultrasonic covert channels using software-defined radio techniques

Robert Sherry
Ethan Bayne
David McLuskie

Sherry, R., Bayne, E. & McLuskie, D. (2023) 'Near-ultrasonic covert channels using software-defined radio techniques'. In: C. Onwubiko, P. Rosati, A. Rege, A. Erola, X. Bellekens, H. Hindy & M.G. Jaatun (eds.) *Proceedings of the International Conference on Cybersecurity, Situational Awareness and Social Media*. Springer, Singapore, International Conference on Cybersecurity, Situational Awareness and Social Media, Cardiff, United Kingdom, 20-21 June 2022.

Near-ultrasonic covert channels using software-defined radio techniques

R. Sherry, E. Bayne, D. McLuskie

Abstract Traditional cybersecurity practices rely on computers only communicating through well-defined expected channels. If malware was developed to use covert channels, such as one created using ultrasonic sound, then this could bypass certain security measures found in computer networks. This paper aims to demonstrate the viability of acoustic covert channels by creating a low-bandwidth ultrasonic frequency channel utilising software-defined radio (SDR) techniques.

Previous work was evaluated to identify the strengths and weaknesses of their implementations. Software-defined radio techniques were then applied to improve the performance and reliability of the acoustic covert channel. The proposed implementation was then evaluated over a range of hardware and compared to previous implantations based on the attributes of their throughput, range, and reliability.

The outcome of this research was an ultrasonic covert channel implemented in GNU Radio. The proposed implementation was found to provide ~47% higher throughput than previous work while using less signal bandwidth. Utilising software-defined radio techniques improves the performance of the acoustic covert channels over previous implementations. It is expected that this technique would be effective in an office environment, but less effective in high security or server environments due to the lack of audio equipment available in these spaces.

Key words: ultrasonic data transfer, covert channel, software-defined radio, SDR, GNU radio

R. Sherry
Division of Cybersecurity, Abertay University, Dundee, DD1 1HG, e-mail: robert@sherry.scot

E. Bayne
Division of Cybersecurity, Abertay University, Dundee, DD1 1HG, e-mail: e.bayne@abertay.ac.uk

D. McLuskie
Division of Cybersecurity, Abertay University, Dundee, DD1 1HG, e-mail: d.mcluskie@abertay.ac.uk

1 Introduction

When a malicious actor attacks a computer system, often their goal is to obtain and extract data. Compromised data can be sold on the black market or leveraged for strategic and economic advantages. The United States Council of Economic Advisers (CEA) [1] estimated that malicious cyber activities cost the US economy between 57 and 109 billion US dollars in 2016 alone.

The large reputational and economic cost of cyberattacks has caused businesses to take cybersecurity much more seriously. However, as cybersecurity is an adversarial field, this will naturally lead to attackers using more advanced techniques in attempts to successfully infiltrate networks.

Lampson [2], in the first recorded work on covert channels, defines covert channels as a ”communication channel that is not intended for information transfer at all”. Covert channels could be used to extract data from air-gapped networks—where a device is physically separated from the network to prevent leakage of valuable data. The ability to cross air-gapped networks has been seen in attacks such as Stuxnet, where malware was spread through non-network means to air-gapped SCADA operator systems. However, malware would not be able to perform data extraction without the use of covert channels.

Beyond bridging air-gapped networks, covert channel attacks have the potential to be an effective attack vector due to the lack of consideration of covert channels in standard security policies. Acoustic covert channels have the potential to be particularly effective due to the common existence of channel requirements (i.e. a speaker and a microphone), and their potential for high rates of data transfer.

The use of a covert channel in an attack has the potential to be particularly devastating, as it makes discovery and incident response considerably more difficult as command and control signalling and data exfiltration could be orchestrated without any data being sent over a monitored network. This would allow it to avoid communication traces and logs being left on network monitor hardware, such as intrusion detection systems (IDS).

This paper aims to improve on previous work in the field by utilising software-defined radio techniques to create a high-throughput method of transferring data utilising a near-ultrasonic audio spectrum. Previous work in this area has neglected to use modern digital signal processing techniques, which limits the potential range and throughput of the demonstrated solutions of the work. Therefore, it can be hypothesized that by utilising modern software-defined radio techniques and digital signal processing techniques, the performance of an acoustic covert channel can be increased.

In this paper, we present three main research contributions: Presenting a novel acoustic covert channel approach to transferring data. Evaluating the proposed model to previous covert channel work. Examining the threats that an acoustic covert channel poses to traditional and high-security networks.

The background section will explore the digital signal processing techniques required to create a communication channel and introduce GNU Radio [3]—the software used to present the proposed SDR approach. The section will then present related work in covert data transfer. The methodology section will outline the development process and testing method used to establish the performance of the covert channel. The results section will summarise performance testing, and the discussion section will analyse results from the proposed method, and compare them to previous work. Finally, a conclusion will be made from the research findings.

2 Background

In the first recorded work on covert channels, Lampson discusses the difficulty of restricting a program's ability to transmit information due to covert channels. These channels have the potential to bypass security measures implemented on a system due to the difficulty in considering all potential avenues in its design.

For an acoustic channel to be covert, the frequency of sound it uses must be inaudible over ambient noise. The minimum frequency that a human can hear varies significantly with age, however, this study will specifically target the adult (over 18 years old) hearing range. For this age group, sensitivity begins to drop off at around 16 kHz, with frequencies over 20 kHz being imperceptible to average ambient background noise [4].

However, there is also an upper limit to the frequency that can be used by a standard computer. The maximum frequency that can be reproduced by a piece of equipment is given by half of its sample rate. This is known as the "Nyquist Limit". Most audio equipment in commodity computers operates at 48KHz, therefore, the Nyquist limit is at 24 kHz.

Acoustic covert channels have been studied by prior literature. Acoustic covert channels only require commonplace hardware to construct—a speaker and microphone. A study in 2018 demonstrated that by using low-level sound card driver functions, an output device could be used as an input device. A sound wave hitting a speaker induces a current much like it would being picked up by a microphone. This allows an acoustic covert channel to be constructed even in higher security environments where microphones are prohibited.

Three main properties can be varied in a wave to encode data—these are phase, frequency, and amplitude. For example, a signal at 19 kHz could represent a "0" symbol, and a signal at 20000 Hz could represent a "1" symbol. This type of modulation is known as "binary frequency shift keying" (BFSK) and is the basis for most previous work in this field.

BFSK has the issue of frequency changes occurring when the amplitude of the wave is not zero. This causes spectral distortion due to the sudden change in frequency, as seen in Figure 1), which can extend far beyond the original transmission frequency and can cause audible artefacts throughout the transmission. These audible artefacts greatly reduce the covertness of the channel, can interfere with adjacent channels, and decrease reliability.

Minimum-shift keying (MSK), illustrated in Figure 2, improves upon BFSK by ensuring that the frequencies are picked so that the frequency changes occur when the amplitude of the wave is zero. MSK effectively solves this issue as the frequency changes do not cause any sudden change in amplitude, which maximises the signal to noise ratio and maintains covertness.

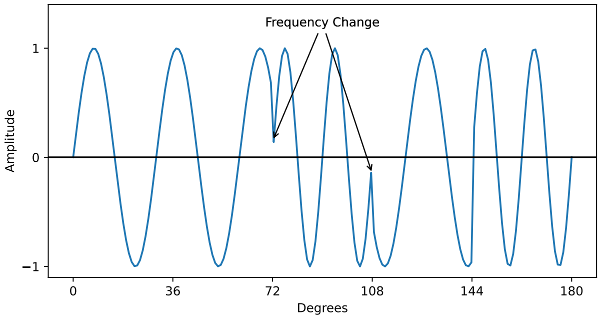


Fig. 1: BFSK signal

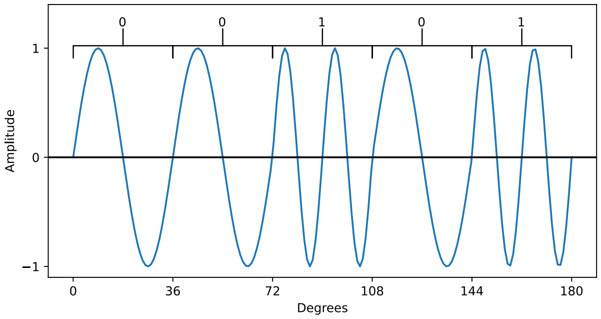


Fig. 2: MSK signal

Modulation is the process of transmitting data, and in the case of an acoustic covert channel is a simple process of turning binary data into a set of tones that can be emitted by the speaker. However, performing demodulation – turning the audio back to binary data – is often significantly more complex due to three key factors; (i) the environment will add noise to the transmission, (ii) the transmissions are received asynchronously, and (iii) demodulation can never be considered fully reliable and may produce invalid data.

These three issues affect all forms of wireless communication, and as such, significant research on how to optimally recover the original signal is available. As the baud rate (the rate at which symbols are transmitted) is increased, the accuracy of each of these steps becomes more important to recover a transmission successfully.

The algorithmic and mathematical complexity of the algorithm to optimally demodulate a signal is very high and previous literature has failed to implement modern techniques to recover the signal, thus significantly limiting the maximum transmission speed and reliability.

However, traditional electromagnetic communication and sound are both waveforms, and hence mathematically equivalent. This means it is possible to utilise existing radio projects that have already implemented algorithms for the acoustic medium. The program this paper utilises is called GNU Radio [3], which is a free and open-source software development toolkit that provides signal processing blocks to implement software-defined radios. These blocks can be linked together in a flowchart to modulate and demodulate a signal.

Additionally, it provides a means to interface with the operating system, allowing access to important functions such as sockets and sound devices, as well as having a large community which have developed a wide range of custom open-source blocks for the program to extend its functionality. While this is designed for use with software-defined radios, it can also be leveraged to create an acoustic modem.

2.1 Related Work

At the date of writing this paper, there are six papers on acoustic covert channels [5, 6, 7, 8, 9, 10]. All of them follow a similar process of modulating data utilising binary frequency shift keying and demodulating the data by cross-correlating the signal with a known preamble and demodulating the data by looking at the magnitude of different frequencies over a static number of samples.

Hanspanch and Goetz [5] repurpose a communication system originally developed for robust underwater communication to create a mesh network of computers utilising common computer audio hardware. They evaluate the range of a single link and its performance over multiple hops. The results from this study show that the approach is very limited, achieving only 20 bit/s of throughput and having 6 seconds of latency per hop, which severely limits the utility of the method as a means to transfer moderately large corpora of data.

Deshotels [6] presents work on acoustic covert channels between mobile devices utilising ultrasonic frequencies. During testing, Deshotels was able to demonstrate a throughput of 345 bit/s between two mobile devices placed back to back. This is the highest throughput backed by empirical evidence in covert channel literature. The authors note that above this speed, they encountered audible clicking that is likely caused by sudden changes in frequency during modulation. Below this baud rate, they solved this issue with pulse shaping to minimise the amplitude of the waves when they shift frequency. The solution presented by Deshotels is sub-optimal as it does not scale to higher baud rates and reduces the signal to noise ratio of the channel, which will affect the reliability at higher ranges and throughputs.

Carrara [7] provides a basis for measuring the covertness of covert channels, with a section that focuses on the optimisation and utility of an acoustic covert channel. The testing presented is in-depth, covering 10 different devices, and analysing multiple channel characteristics that consider the background noise, reverberation, and frequency responses of the devices that were used for testing. Channel parameters are also tested in-depth and optimised for performance, however, this data does not appear to inform subsequent tests, which may limit the maximum throughput possible with the proposed model. While the individual channel implemented by Carrara is fairly low performance, by combining multiple, they can achieve 230 bit/s within the ultrasonic range.

Wong [8] discusses the design and implementation of an ultrasonic covert channel, experimenting with different modulation schemes within the acoustic medium. They validate their results on a wide range of hardware and measure the throughput and reliability at different distances. Wong's data is used to analyse the effectiveness of an acoustic covert channel and discuss its limitations. While the results of Wong's paper seemed initially promising, their analyse is not particularly thorough. The throughput of 500 bit/s utilising Quaternary Frequency Shift Keying (QFSK) is stated throughout the paper, however, it appears to only have been explored theoretically and is not experimentally validated other than demonstrating that there is enough bandwidth for it to fit in the ultrasonic spectrum.

Wong also claims the discovery of utilising pulse shaping to eliminate the issues of clicking found in previous work, however, in the same paper cited as having this issue, the same solution is both proposed and implemented. Additionally, the throughput measurements provided by Wong neglect to include the overhead from packetization and error loss, meaning that their performance data is inflated and would not be representative of its effectiveness. Wong's paper was initially the basis for this research, however, when the work was reproduced, the synchronisation technique began to fail at over 50 bits per second on high-end audio hardware, which does not support the results provided in Wong's paper.

Smye [9] investigates ultrasound as a communication method, commenting on potential security issues with their use. They note several existing frameworks which claim to provide high throughput over an ultrasonic channel. The authors note that the signal strength significantly degrades with realistic test conditions. Smye's project documents a similar independent implementation of this research using GNU Radio, with the author achieving a throughput of 50 bit/s over 2m.

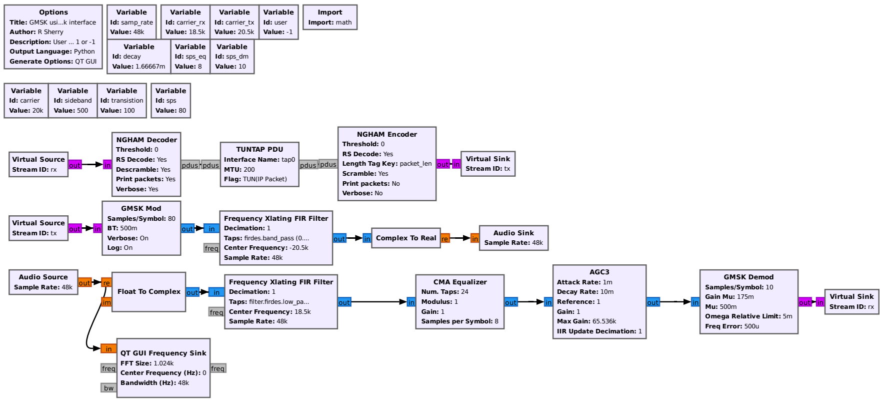
Zarandy et al. [10] investigate the use of ultrasound signals as a censorship-resistant mesh network and covid contract tracing physical layer between mobile devices. They demonstrated throughput of 685.7 bit/s and a maximum range of 8m using an 8-PSK modulation method. The limiting factor in the Zarandy et al. study was the overhead from the error correction required to maintain a reliable connection and correct for carrier phase drift.

3 Methodology

To validate and build upon previous work done within this research domain, GNU Radio [3] is used to create an audio signal processor that can generate and decode sound signals sent in the near-ultrasonic frequency range. GNU Radio is a free and open-source toolkit which is used to rapidly develop a signal-processing system. GNU Radio is traditionally used for SDR systems, however, in this paper, it is used to process sound signals to and from data streams.

3.1 Implementation

Figure 3 shows the final and functional implementation of the duplex ultrasonic covert channel in GNU Radio. This includes packetization, synchronisation, equalisation, and automatic gain control which all work to maximise the reliability of the channel. This is accessed through a Linux TAP virtual network interface, which allows for full-duplex Transfer Control Protocol (TCP) communication across the link by virtualising a layer 3 ethernet interface.



Options
Title: GMSK usi...k interface
Author: R Sherry
Description: User ... 1 or -1
Output Language: Python
Generate Options: QT GUI

Variable **Id:** samp_rate **Value:** 48k
Variable **Id:** carrier_rx **Value:** 18.5k
Variable **Id:** carrier_tx **Value:** 20.5k
Variable **Id:** user **Value:** -1
Import **Import:** math
Variable **Id:** decay **Value:** 1.66667m
Variable **Id:** sps_eq **Value:** 8
Variable **Id:** sps_dm **Value:** 10
Variable **Id:** carrier **Value:** 20k
Variable **Id:** sideband **Value:** 500
Variable **Id:** transistion **Value:** 100
Variable **Id:** sps **Value:** 80

Virtual Source **Stream ID:** rx

NGHAM Decoder
Threshold: 0
RS Decode: Yes
Descramble: Yes
Print packets: Yes
Verbose: Yes

TUNTAP PDU
Interface Name: tap0
MTU: 200
Flag: TUN(IP Packet)

NGHAM Encoder
Threshold: 0
RS Decode: Yes
Length Tag Key: packet_len
Scramble: Yes
Print packets: No
Verbose: No

Virtual Sink **Stream ID:** tx

Virtual Source **Stream ID:** tx

GMSK Mod
Samples/Symbol: 80
BT: 500m
Verbose: On
Log: On

Frequency Xlating FIR Filter
Decimation: 1
Taps: firdes.band_pass (0....
Center Frequency: -20.5k
Sample Rate: 48k

Complex To Real

Audio Sink **Sample Rate:** 48k

Audio Source **Sample Rate:** 48k

Float To Complex

Frequency Xlating FIR Filter
Decimation: 1
Taps: filter.firdes.low_pa...
Center Frequency: 18.5k
Sample Rate: 48k

CMA Equalizer
Num. Taps: 24
Modulus: 1
Gain: 1
Samples per Symbol: 8

AGC3
Attack Rate: 1m
Decay Rate: 10m
Reference: 1
Gain: 1
Max Gain: 65.536k
IIR Update Decimation: 1

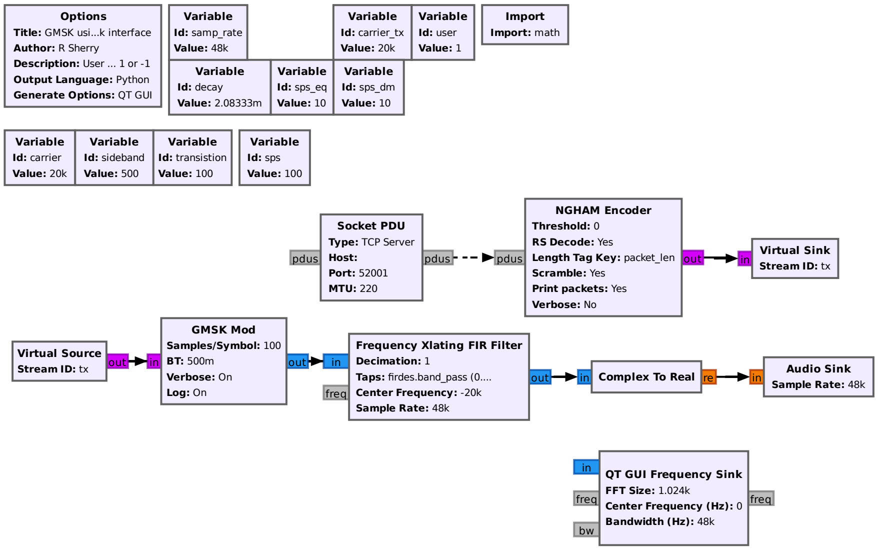
GMSK Demod
Samples/Symbol: 10
Gain Mu: 175m
Mu: 500m
Omega Relative Limit: 5m
Freq Error: 500u

Virtual Sink **Stream ID:** rx

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 48k

Fig. 3: Final flowgraph utilising a duplex Linux TAP interface

Figures 4 and 5 show the flowgraphs that were used to explicitly test the covert channel in the experiments presented in this research. These flowgraphs are stripped-down versions of the full model presented in figure 3 that use the channel in a simplex configuration, with communication accessed through a socket server. These flowgraphs were used to establish the throughput and error rate of a single link and eliminate additional errors caused by the additional channel used for packet acknowledgement.



Options
Title: GMSK usi...k interface
Author: R Sherry
Description: User ... 1 or -1
Output Language: Python
Generate Options: QT GUI

Variable **Id:** samp_rate **Value:** 48k
Variable **Id:** carrier_tx **Value:** 20k
Variable **Id:** user **Value:** 1
Import **Import:** math
Variable **Id:** decay **Value:** 2.08333m
Variable **Id:** sps_eq **Value:** 10
Variable **Id:** sps_dm **Value:** 10
Variable **Id:** carrier **Value:** 20k
Variable **Id:** sideband **Value:** 500
Variable **Id:** transistion **Value:** 100
Variable **Id:** sps **Value:** 100

Socket PDU
Type: TCP Server
Host:
Port: 52001
MTU: 220

NGHAM Encoder
Threshold: 0
RS Decode: Yes
Length Tag Key: packet_len
Scramble: Yes
Print packets: Yes
Verbose: No

Virtual Sink
Stream ID: tx

Virtual Source
Stream ID: tx

GMSK Mod
Samples/Symbol: 100
BT: 500m
Verbose: On
Log: On

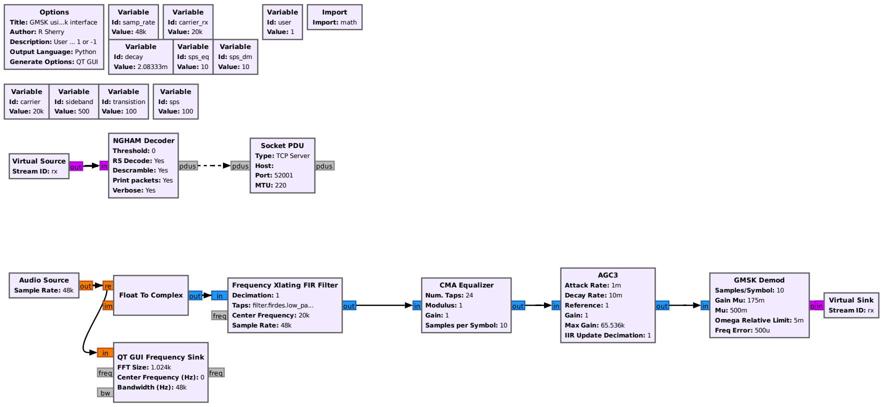
Frequency Xlating FIR Filter
Decimation: 1
Taps: firdes.band_pass (0....
Center Frequency: -20k
Sample Rate: 48k

Complex To Real

Audio Sink
Sample Rate: 48k

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 48k

Fig. 4: Transmission flowgraph



Options
Title: GMSK usi...k interface
Author: R Sherry
Description: User ... 1 or -1
Output Language: Python
Generate Options: QT GUI

Variable **Id:** samp_rate **Value:** 48k
Variable **Id:** carrier_rx **Value:** 20k
Variable **Id:** user **Value:** 1
Import **Import:** math
Variable **Id:** decay **Value:** 2.08333m
Variable **Id:** sps_eq **Value:** 10
Variable **Id:** sps_dm **Value:** 10
Variable **Id:** carrier **Value:** 20k
Variable **Id:** sideband **Value:** 500
Variable **Id:** transistion **Value:** 100
Variable **Id:** sps **Value:** 100

Virtual Source
Stream ID: rx

NGHAM Decoder
Threshold: 0
RS Decode: Yes
Descramble: Yes
Print packets: Yes
Verbose: Yes

Socket PDU
Type: TCP Server
Host:
Port: 52001
MTU: 220

Audio Source
Sample Rate: 48k

Float To Complex

Frequency Xlating FIR Filter
Decimation: 1
Taps: filter.firdes.low_pa...
Center Frequency: 20k
Sample Rate: 48k

CMA Equalizer
Num. Taps: 24
Modulus: 1
Gain: 1
Samples per Symbol: 10

AGC3
Attack Rate: 1m
Decay Rate: 10m
Reference: 1
Gain: 1
Max Gain: 65.536k
IIR Update Decimation: 1

GMSK Demod
Samples/Symbol: 10
Gain Mu: 175m
Mu: 500m
Omega Relative Limit: 5m
Freq Error: 500u

Virtual Sink
Stream ID: rx

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 48k

Fig. 5: Receiving flowgraph

The key components of these flowgraphs that allows the model to be used for acoustic data transfer are the “Frequency Xlating FIR Filter” and the “Complex to Real” blocks. The former allows the signal to be shifted from baseband (centred around 0 Hz) to ultrasonic frequencies. This is intended to extract a signal and bring it down to baseband, however, by providing it with a negative frequency it can be used to do the reverse. The latter converts the complex samples to real samples that audio devices can understand.

The NGHAM Encoder/Decoder blocks [11] are an open-source addition to GNU-Radio which provides a robust scheme for forwarding error correction and packetization.

3.2 Channel Bandwidth

The bandwidth of the channel was determined under GNU Radio utilising a model. This parameter is important to know as it determines the number of channels that can be fit within the near-ultrasonic range. For GMSK this is directly linked to the baud rate, and as such, this value was measured for each baud rate tested in the previous experiment.

This flowgraph will continuously modulate data read in from a file, which is then demodulated with the result being output to a terminal. The “sideband” value was initialised at 100 Hz and increased by 5 Hz till the signal began to demodulate correctly. This value is used to determine the range of frequencies above and below the carrier frequencies that should go through the filter, and as such, is half the total channel bandwidth.

3.3 Testing

A standardised OS test environment containing the project was created using a 32GB USB stick to ensure consistency across the machines used in testing. Each USB stick contained an identical instance of Ubuntu 19.10 with GNU Radio 3.7.14.4 and the NGHam packetization module installed.

3.3.1 Equipment

To determine the viability of this system in a realistic attack scenario where the equipment may be known but out of the control of the attacker, the system was tested over a range of hardware at three different quality points classified as L (low), M (medium), and H (high) quality by the researchers. Every combination of hardware was tested, giving a total of 9 test sets. The hardware used in testing can be found in Table 1.

3.3.2 Test Method

Testing was conducted in an Abertay University computer lab. The experiments were carried out whilst other people were present in the lab. The added noise from people talking created a realistic environment, and the added interference was deemed

Speakers	
Quality	Model
L	Viobyte DH-660 (Headphones)
M	Genius SP-S110
H	Logitech Z370
Microphones	
Quality	Model
L	Generic Office Headset (no brand)
M	Tonor TN120072BL
H	Blue Yeti Nano

Table 1: Equipment used for experiments

unlikely to affect results due to being far outside the expected audio transmission range.

A test file was generated by reading 10,800 bytes of random data from "/dev/urandom", which was reused for each test and verifiable. A test harness was built on top of GNU Radio's python processes to automate testing and take measurements of speed and error rate.

A checksum was calculated at each side of the transmission to validate the integrity of received packets. As such the reliability of the link can be calculated as follows:

$$\frac{bytesreceived}{10,800} \times 100 = reliabilitypercentage$$

Effective throughput was calculated as such:

$$\frac{bytesreceived}{timetaken}$$

The speaker and microphone were placed 3, 6 and 9 meters apart on opposing office chairs using a measuring tape for all combinations of equipment (Figure 6). The transmitting computer was configured to 80% volume, and to play audio only through the left channel. The exception to this was the headphones, which were set to the system's max volume level of 150%, as the power is significantly lower than a speaker system. The headphone left cup was placed over the top of the right cup to prevent it from physically interfering with the signal. Without these alterations, the reliability of the channel using headphones was too low at the tested ranges.

Each combination of audio hardware was tested at three baud rates – 480, 600, and 800 baud. If any baud rate failed at a previous range, they were not tested at subsequent ranges as it was assumed the data would continue to be irrecoverable. While this assumption may not necessarily always be true, access to perform extended experiments was limited due to time restrictions.

Testing was repeated at three different frequencies – 18 kHz, 20 kHz, and 22 kHz – to eliminate any issues caused by the individual frequency response curves of the equipment. Each test was only conducted once, again due to limited lab access due to time restrictions.

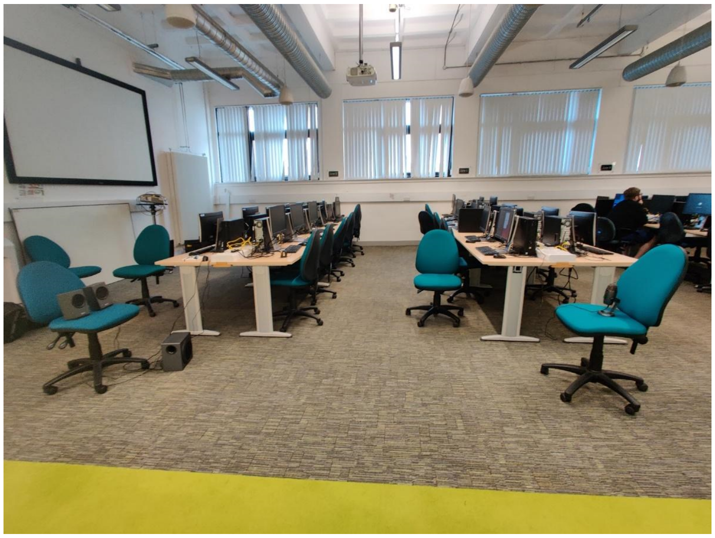


Fig. 6: Lab experiment setup

Overall, 9 combinations of hardware were tested with the devised covert channel over 3 distances, 3 baud-rates and 3 frequencies each—providing 243 tests total.

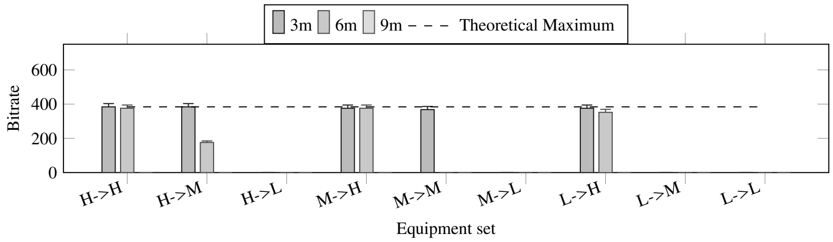
4 Results

4.1 Results at 18 kHz

18 kHz provided very good performance overall (Figure 7). While as the range and baud rate increased, the error rates for many of the devices increased, only the lowest quality microphone had issues receiving signals for demodulation.

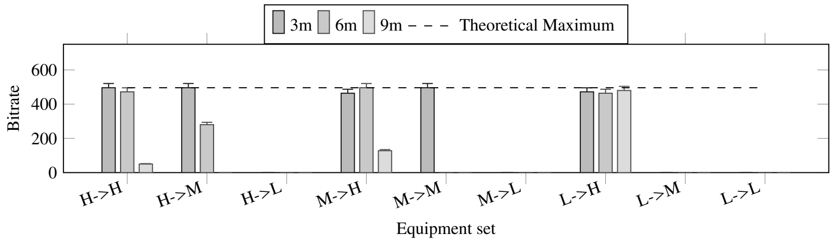
4.2 Results at 20 kHz

The performance at 20 kHz is considerably lower, with only the higher-end devices able to successfully decode the data at higher baud rates and ranges (Figure 8). However, at the lower range of 3m, and a baud rate of 480, many of the devices are still able to consistently demodulate a signal.



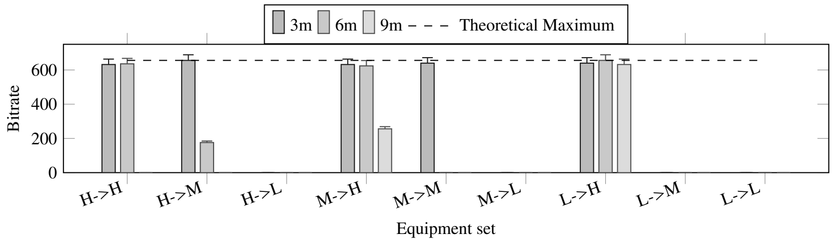
Error rate (%)									
	H->H	H->M	H->L	M->H	M->M	M->L	L->H	L->M	L->L
3	0	0	100	2.08	4.17	100	2.08	100	100
6	2.08	54.17	100	2.08	100	100	8.33	100	100
9	100	100	100	100	100	100	100	100	100

(a) 18 kHz at 480 Baud



Error rate (%)									
	H->H	H->M	H->L	M->H	M->M	M->L	L->H	L->M	L->L
3	0	0	100	6.45	0	100	4.84	100	100
6	4.84	43.55	100	0	100	100	6.45	100	100
9	89.92	100	100	74.19	100	100	3.23	100	100

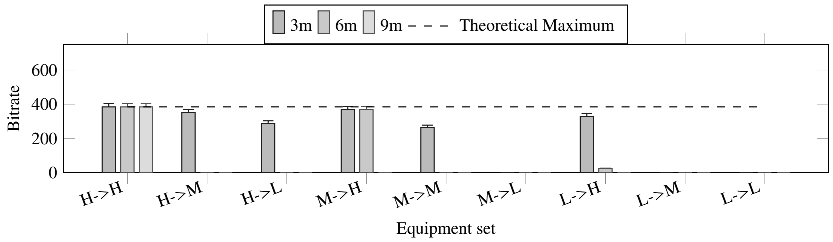
(b) 18 kHz at 600 Baud



Error rate (%)									
	H->H	H->M	H->L	M->H	M->M	M->L	L->H	L->M	L->L
3	3.66	0	100	3.66	2.44	100	2.44	100	100
6	3.66	73.17	100	4.88	100	100	0	100	100
9	100	100	100	60.98	100	100	3.66	100	100

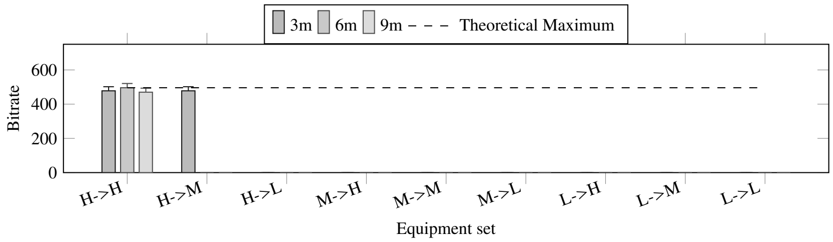
(c) 18 kHz at 800 Baud

Fig. 7: Effective throughput 18 kHz with 95% confidence intervals



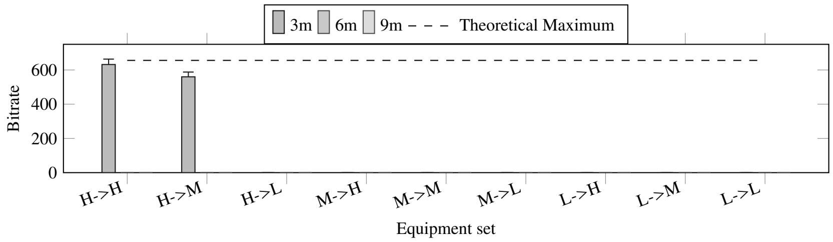
	Error rate (%)								
	H->H	H->M	H->L	M->H	M->M	M->L	L->H	L->M	L->L
3	0	8.33	25	4.17	31.25	100	14.58	100	100
6	0	100	100	4.17	100	100	93.75	100	100
9	0	100	100	100	100	100	100	100	100

(a) 20 kHz at 480 Baud



	Error rate (%)								
	H->H	H->M	H->L	M->H	M->M	M->L	L->H	L->M	L->L
3	3.63	3.63	100	100	100	100	100	100	100
6	0	100	100	100	100	100	100	100	100
9	5.24	100	100	100	100	100	100	100	100

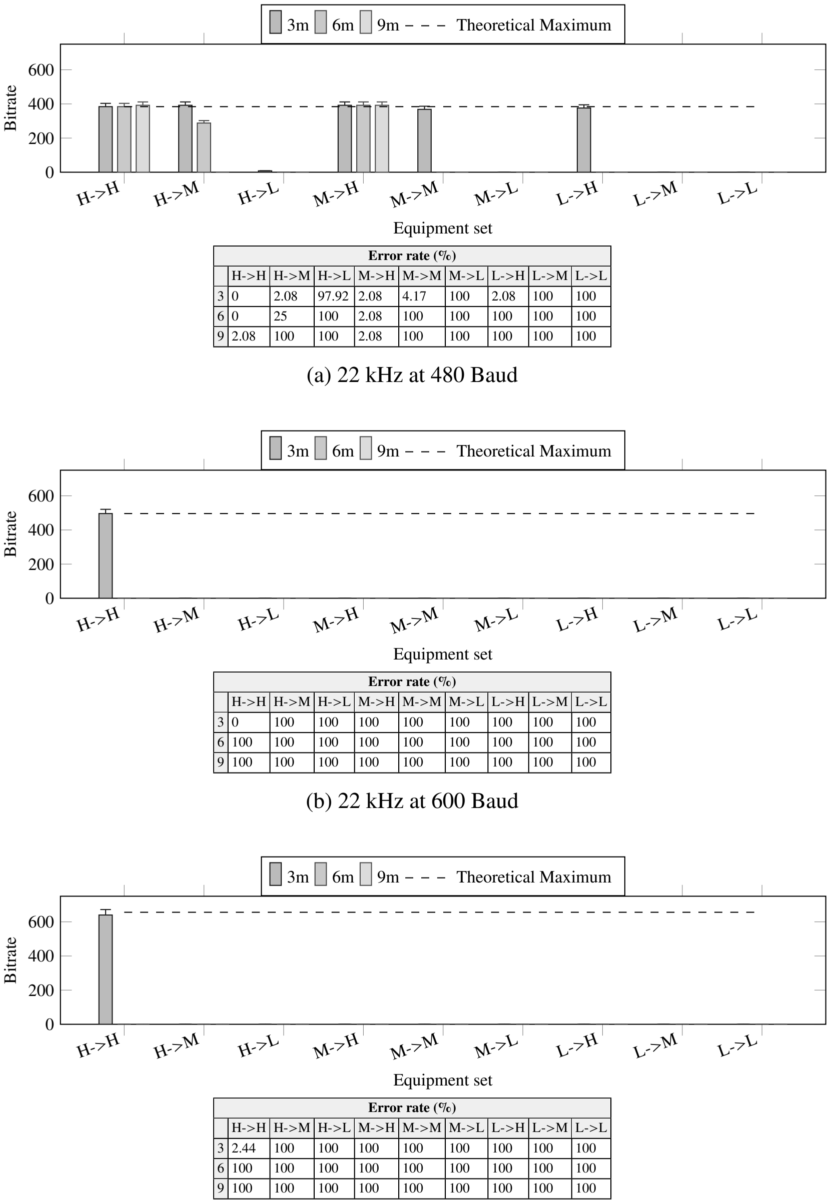
(b) 20 kHz at 600 Baud



	Error rate (%)								
	H->H	H->M	H->L	M->H	M->M	M->L	L->H	L->M	L->L
3	3.66	14.63	100	100	100	100	100	100	100
6	100	100	100	100	100	100	100	100	100
9	100	100	100	100	100	100	100	100	100

(c) 20 kHz at 800 Baud

Fig. 8: Effective throughput 20 kHz with 95% confidence intervals



Error rate (%)									
	H->H	H->M	H->L	M->H	M->M	M->L	L->H	L->M	L->L
3	0	2.08	97.92	2.08	4.17	100	2.08	100	100
6	0	25	100	2.08	100	100	100	100	100
9	2.08	100	100	2.08	100	100	100	100	100

(a) 22 kHz at 480 Baud

Error rate (%)									
	H->H	H->M	H->L	M->H	M->M	M->L	L->H	L->M	L->L
3	0	100	100	100	100	100	100	100	100
6	100	100	100	100	100	100	100	100	100
9	100	100	100	100	100	100	100	100	100

(b) 22 kHz at 600 Baud

Error rate (%)									
	H->H	H->M	H->L	M->H	M->M	M->L	L->H	L->M	L->L
3	2.44	100	100	100	100	100	100	100	100
6	100	100	100	100	100	100	100	100	100
9	100	100	100	100	100	100	100	100	100

(c) 22 kHz at 800 Baud

Fig. 9: Effective throughput 22 kHz with 95% confidence intervals

4.3 Results at 22 kHz

Performance at 22 kHz is very weak, with only the highest-end equipment able to receive data at high ranges and throughputs (Figure 9). However, at the lower baud rate of 480, the high-end microphone was able to receive data at short ranges from all three speakers.

5 Discussion

Testing showed that transmission was possible at all three baud rates of 480, 600, and 800 baud, but performance varied significantly with the frequency and range of communication, as well as the quality of the equipment.

Table 2 demonstrates that as the baud rate increases, so does the bandwidth required for the channel to operate, decreasing the number of channels that could be implemented in the near ultrasonic range from 18 kHz to 24 kHz.

Baud Rate	Channel Bandwidth (Hz)	No. of Channels
480	440	14
600	560	11
800	740	8

Table 2: Sideband Measurement Results

Communication at 18 kHz was by far the strongest demonstrated by these experiments, with communication being demonstrated as feasible in most tested configurations. Communication at 20 and 22 kHz drops off considerably, with only the highest-end equipment being able to reliably transmit data at this frequency. However, at close ranges and the lowest baud rate, the transmission was still shown to be viable at this frequency range. This is likely due to both the speaker and microphone having a poor frequency response at high frequencies, as distortion in inaudible frequencies is unlikely to be noticed by most users.

5.1 Input Devices

The low-end microphone was almost completely unable to demodulate data consistently due to the error rate being too high. The low-end microphone was only considered usable in one experiment—480 Baud, 20 kHz, at a 3m distance. This is likely due to both the frequency and distance of the audio source being outside of the device's design parameters.

The midrange microphone was able to decode data successfully at short ranges, however, error rates became unusable at distances beyond 6m. This is once again likely due to its design and intended use case, which is made for employees gathered around a table.

The high-end microphone performed extremely well, being able to demodulate signals at most ranges, baud rates and frequencies. This is likely due to the high fidelity of this equipment, which greatly helps with signal recovery.

5.2 Output Devices

The low-end headphones had surprisingly high performance, with solid reliability rates at all three baud rates when paired with a high-end microphone at 18 kHz. This is likely due to having decent audio fidelity, but poor volume, requiring the sensitivity of the high-end microphone to recover the signal.

The mid-range speakers had overall poor performance. This is potentially due to the amplifier having poor fidelity at ultrasonic frequencies, causing error rates to increase significantly. This can be established due to the high-end microphone's uncharacteristically poor performance when communicating with this device.

The high-end speakers performed very well, likely due to a combination of high fidelity at high volume, providing a best-case scenario for the receiver. This kind of speaker is unlikely to be found on most office employee computers, but similar quality audio equipment could exist in conference rooms or studio environments.

5.3 Observations

The quality of the microphone used appeared to have the greatest effect on performance, with the lowest quality microphone consistently having the worst performance, and the highest quality microphone having the best performance.

The headset microphone is designed to be very close to the user to operate effectively, whereas the other two microphones tested are cardioid microphones, designed to pick up audio from a distance. As such, similarly designed cardioid microphones will likely be effective at comparable ranges. Additionally, the headset microphone used in this testing was of particularly poor quality, with significant noise in the base signal. This was done to represent an absolute worst-case scenario.

As such, further testing with the headset microphone could be conducted at closer ranges ($<$1m), to demonstrate an audio device being placed nearby an infected computer, as this is likely a much more representative scenario for these types of devices.

In scenarios where only simplex communication is desired, the attacker is entirely in control of the quality of the receiving microphone. As such, a higher-end microphone has demonstrated a capability to greatly increased both the range and maximum throughput of the channel, and it is likely a microphone designed for higher frequencies, or utilising a higher sample rate, could increase the proposed method's performance further.

The effective throughput of each baud rate after the overhead added by packetization was found to be 384 bit/s, 496 bit/s, and 656 bit/s for 480, 600 and 800 baud respectively, which shows significant overhead in the packetization and error correction added to the signal, however, this was necessary for reliable communication to take place.

A summary of performance results from this study is compared with related work in Table 3.

Deshotels's work [6], which was tested at effectively 0m distance with mid-range audio equipment, provides comparable results to the proposed method from this research at a distance of 3m at 18 kHz using mid-range equipment. In this case, high reliability was demonstrated at 656 bit/s, and a bandwidth of 740 Hz. Deshotels tests a much lower throughput link of 8.61 bit/s at up to 100 ft. This work only tests up to a range of 9 meters. For a full comparison to be made between the two methods, greater distances must be tested.

Deshotels's work [6] references an issue with clicking during transmission, which is solved in this work by using minimum shift keying as opposed to pulse shaping. This allows for link speeds higher than the 345 bit/s while maintaining a higher signal to noise ratio. This acted as the performance ceiling for Deshotels's work, and as such, their work could be improved upon by utilising this modulation method.

Zarandy et al. [10] achieve an effective throughput of 685.7 bit/s and claim that up to a range of 8m would be possible with an unknown effect on throughput. The use of a complex 8-PSK modulation scheme means that significant error correction was required for both data recovery and carrier-phase drift tracking. Given that transmission quality dropped off significantly past 6m it is likely that further error correction, or a drop-in baud rate, would be required to continue communication, further limiting the effective throughput. This makes it slightly faster than our 800 baud implementation which provides 656 bit/s; however, the author notes an audible clicking due to the spectral distortion caused by the modulation phase changes, making it unsuitable as a covert channel. Given the complexity of the modulation scheme, it is also likely

Author	**Throughput (bit/s)**	**Channel Bandwidth (Hz)**	**Range (m)**
Sherry, Bayne & McLuskie (2021)	384	440	3-9m
	496	560	3-6m
	656	740	3-6m
Hanspach & Goetz (2014) [5]	20	1000	1-8.2m
Deshotels (2014) [6]	345	1000	0m
	8.61	1000	30.48m
Carrara (2015) [7]	230	500	1-11m*
Wong (2018) [8]	180	355	2-4m
Zarandy et al. (2020) [10]	685.7	undisclosed	0-8m

Table 3: Comparison to previous work

that lower quality equipment would struggle with the precision required to carry the signal.

Wong's work [8] demonstrated an effective throughput of around 180 bit/s at up to 2m, with significant reliability issues above this distance. Compared to Wong's approach, the solution proposed by this research demonstrates a higher effective throughput of 384 bit/s at a comparable bandwidth of 440 Hz, with a much greater range of up to 9m on certain equipment. Wong also references a theoretical system utilising QFSK to provide a baud rate of 500, with a bandwidth of above 1,140 Hz. This would still be a lower performance than the results achieved in this paper at both 600 and 800 baud at up to 6m.

Carrara's work [7] demonstrates a maximum throughput of 230 bit/s utilising a bandwidth of 500 Hz, at up to distances of 11 m, with low bit error rates. This paper demonstrates 384 bit/s, utilising a bandwidth of 440 Hz, at distances of up to 9m, with low bit error rates on mid to high-end equipment. This is higher performance but also utilises a much simpler modulation system. Carrara achieves these data rates by combining multiple carriers at a much lower throughput utilising a scheme called OFDM, as opposed to this system which increases the raw baud rate on a single carrier.

The performance outlined in this research could be improved by utilising multiple reliable carriers at 480 baud or lower to reliably scale performance up even higher. GNU Radio has features to support this type of implementation, however, the lack of documentation and the complexity of the modules pushed it out of the scope of this work.

Hanspach and Goetz [5] demonstrate a maximum throughput of 20 bit/s at a range of up to 9m, utilising a bandwidth of approximately 1 kHz. This is considerably lower performance than the other research discussed. However, the utilisation of a mesh network is a novel idea and could be combined to produce higher throughput, low range links, to allow reliable communication over a considerable distance.

As this work demonstrates higher performance regarding bandwidth and throughput to previous implementations, utilising software-defined radio techniques does provide a substantial performance benefit compared to previous research. This work demonstrates a maximum performance of 656 bit/s at ranges under 6m, and throughput of up to 384 bit/s at ranges of up to 9m, with the potential for this to work at longer ranges. Additionally, the issue of clicking shown in other research is solved by utilising minimum shift keying which eliminates phase discontinuities at frequency changes and allows for higher covert data rates without affecting the signal to noise ratio of the transmission.

GNU Radio also provides lower computational complexity than previous work, which uses various variants of the fast Fourier transform (FFT) algorithm which are relatively complex to calculate.

The proposed model operates at lower bandwidths than previous work, allowing multiple channels to potentially be combined to increase performance further, or be utilised to link multiple computers together with separate exclusive channels, to extend the range.

This type of covert channel poses a high risk to traditional networks, as this type of threat is unlikely to be considered in their security model. This technique is most likely to be effective in an open-plan office, as many of the computers will have microphones and speakers attached to them.

However, this technique may not be effective in scenarios where the infected computer is a server, as server hardware is considerably less likely to contain a speaker for communication and are very unlikely to have a microphone for duplex communication. However, if it is connected to an interface device, such as a KVM console, it may still be possible to occasionally transmit data.

In the case of a high-security environment, this technique is unlikely to be effective as microphones or personal devices are usually not allowed in such environments. Although other work has demonstrated the ability to utilise speakers as microphones, this would require a direct line of sight and would be unlikely to work at the baud rates demonstrated in this research. However, this technique could still be utilised to transmit data between applications running on a computer which could potentially bypass security restrictions.

5.4 Countermeasures

There are broadly three categories of countermeasures that could be taken against this type of attack. The first one would be to filter both input and output devices to prevent them from using inaudible frequencies. This would prevent transmission and could be done through either software or hardware.

The second category of defences would be to jam ultrasonic transmission by constantly transmitting noise on these frequencies. This could only be effective in unoccupied areas such as server rooms, as long term exposure to ultrasonic noise has been found to cause headaches [12]. However, this is unlikely to be particularly effective as if there are no bystanders, then the channel does not need to be restricted to the near ultrasonic range as the channel will remain covert.

The final category would be to detect ultrasonic sounds, which would alert administrators to their use. An effective way to do this would be to distribute monitoring for ultrasonic frequencies between all computers in an office. This is likely to be the most effective solution as it is the most widely applicable and easy to implement while being difficult to bypass due to its distributed nature.

6 Conclusion

To conclude, software-defined radio techniques have been demonstrated to provide significantly higher throughput performance (~47%) through an acoustic covert channel than previous implementations, however, the reliability demonstrated by this research is highly dependent on the carrier frequency, equipment quality, and communication range.

Furthermore, the utilisation of minimum shift keying has been demonstrated to eliminate the issue of clicking found in published work in the field, and the utilisation of free open-source software has allowed for work to be easily reproduced and expanded upon.

The threats to traditional and high-security networks have been analysed, with the threat to open-plan offices identified as particularly high due to the prevalence of audio equipment required for covert channel attacks. The threat to high security or severe environment identified was deemed to be at a lower risk, due to the lack of basic channel requirements required for the covert channel to work.

Suggestions for countermeasures have been made, with the idea of a distributed monitoring system proposed as the most effective. The use of SDR techniques has been proven to be a novel and effective approach to the implementation of a covert channel, and its potential to provide high-performance covert links should not be underestimated.

7 Future Work

Future work could be done in this area by utilising GNU Radio's built-in OFDM blocks to use multiple sub-carriers to expand the throughput, without compromising on reliability or range. Additionally, the low bandwidth of this solution was specifically optimised to allow for multiple channels. This makes it suitable to create a mesh network as demonstrated by Goertz. If a form of carrier access scheme could be added to GNU Radio, this would allow for the range to increase significantly by repeating a signal through multiple infected nodes.

Secondly, significant validation work is still required to fully evaluate the reliability of this channel in a wide range of scenarios. As the microphone has been identified as a primary factor in the reliability of the channel, a wide range of microphones should be tested to determine the overall effectiveness of the technique.

Longer range communication above 9m should also be explored, to determine the maximum range of communication. This could be done at lower baud rates to allow for increased reliability with signal attenuation. This work also has limited statistical validation due to time constraints. The experiments outlined in this research above would ideally be repeated multiple times to ensure validity. Additionally, the testing conducted in this paper was based on long, continuous transmissions. This methodology should also be tested with burst based transmission, where signals are randomly sent at larger intervals, to determine the reliability of the synchronisation method utilised in this paper.

Lastly, further work validating countermeasures against covert channel attacks should be explored in practice.

References

1. The Council of Economic Advisers, “The Cost of Malicious Cyber Activity to the U.S. Economy,” no. February, pp. 1–62, 2018. [Online]. Available: https://www.fbi.gov/investigate/cyber%0Ahttps://www.whitehouse.gov/wp-content/uploads/2018/02/The-Cost-of-Malicious-Cyber-Activity-to-the-U.S.-Economy.pdf
2. B. Lampson, “A note on the confinement problem,” *Communications of the ACM*, vol. 16, no. 10, pp. 613–615, oct 1973. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/362375.362389
3. GNU Radio Project. [Online]. Available: https://www.gnuradio.org/about/
4. K. Ashihara, “Hearing thresholds for pure tones above 16kHz,” *The Journal of the Acoustical Society of America*, vol. 122, no. 3, p. EL52, aug 2007. [Online]. Available: https://asa.scitation.org/doi/abs/10.1121/1.2761883
5. M. Hanspach and M. Goetz, “On Covert Acoustical Mesh Networks in Air,” *Journal of Communications*, vol. 8, no. 11, pp. 758–767, jun 2014. [Online]. Available: https://arxiv.org/abs/1406.1213v1
6. L. Deshotels, “Inaudible sound as a covert channel in mobile devices,” in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*. San Diego, CA: USENIX Association, Aug. 2014. [Online]. Available: https://www.usenix.org/conference/woot14/workshop-program/presentation/deshotels
7. B. Carrara and C. Adams, “On characterizing and measuring out-of-band covert channels,” in *IH and MMSec 2015 - Proceedings of the 2015 ACM Workshop on Information Hiding and Multimedia Security*, 2015, pp. 43–54. [Online]. Available: http://dx.doi.org/10.1145/2756601.2756604
8. W. I. Wong, “Crossing the Air Gap — An Ultrasonic Covert Channel,” Thesis, Royal Military College of Canada, 2018.
9. A. Smye, “An Introduction to Ultrasound Security Research,” NCC Group, Tech. Rep., jul 2020. [Online]. Available: https://research.nccgroup.com/wp-content/uploads/2020/07/ultrasound-whitepaper-v1.0.pdf
10. A. Zarandy, I. Shumailov, and R. Anderson, “BatNet: Data Transmission Between Smartphones Over Ultrasound,” aug 2020. [Online]. Available: https://arxiv.org/pdf/2008.00136.pdf
11. “NGHam.” [Online]. Available: https://github.com/skagmo/ngham
12. B. Smagowska and M. Pawlaczyk-Luszczyńska, “Effects of Ultrasonic Noise on the Human Body—A Bibliographic Review,” *https://doi.org/10.1080/10803548.2013.11076978*, vol. 19, no. 2, pp. 195–202, 2015. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/10803548.2013.11076978