

Covert Data Exfiltration from Smartphones through mmWave FMCW Radar

P. Dhatchaya
2025



Covert Data Exfiltration from Smartphones through mmWave FMCW Radar

Dhatchaya Prabakaran

Index No : 20000413

Supervisor: Dr. Asanka Sayakara

May 2025

Submitted in partial fulfillment of the requirements of the
B.Sc.(Honours) in Computer Science Final Year Project



Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: P. Dhatchaya



Signature of candidate

Date: June 28, 2025

This is to certify that this dissertation is based on the work of Ms. P.Dhatchaya under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Supervisor Name: Dr. Asanka P. Sayakkara



Signature of Supervisor

Date: June 28, 2025

Abstract

This research explores a novel covert communication method using smartphone-generated vibrations detected by millimeter-wave (mmWave) Frequency-Modulated Continuous Wave (FMCW) radar. Traditional covert channels rely on internet connectivity, sound, or proximity-based methods that can be disabled or monitored. In contrast, this work demonstrates a contactless, connectivity-free covert channel using only the smartphone’s vibration motor and mmWave radar, aiming to assess its feasibility and cybersecurity implications.

To achieve this, we developed a custom On-Off Keying (OOK) encoding scheme with a symbol duration of 80ms, with optimized vibration intensity to ensure that vibrations remain imperceptible to users while still being detectable by the radar. The evaluation of our system focused on two main aspects: the stealthiness of the covert channel and the technical feasibility of using mmWave radar to extract data. To assess stealth, qualitative tests with human participants were conducted to verify that the vibration signals remained imperceptible across common surfaces and at reduced amplitude levels. For technical feasibility, Experiments were conducted across five different Google Pixel smartphone models, distances ranging from 0.6m to 3.4m, different encoding schemes, device orientations, and through common household barriers such as cardboard, plastic, and glass. The mmWave radar setup, operating at 77GHz captured vibration signals for decoding.

Results show that the radar achieves a 0% Bit Error Rate (BER) at distances up to 3.1m under line-of-sight conditions. At 3.4m, the BER increases significantly to approximately 56.25%. Experiments involving physical barriers revealed that thin materials allow reliable decoding, while thicker or denser barriers lead to noticeable performance degradation. Despite these limitations, the system maintained a data rate of 12.5bps under optimal conditions, enabling covert data transmission without triggering user suspicion or requiring any connectivity.

These findings confirm that mmWave radar can be repurposed as a covert receiver for vibration-based data transmission. This work contributes to cybersecurity research by identifying and demonstrating a novel covert-channel threat and highlights the urgent need for new security mechanisms that consider non-traditional sensing technologies in threat models.

Preface

This thesis explores the application of Millimetre -Wave (mmWave) radar in the cybersecurity domain, specifically focusing on its potential to detect covert data transmissions via smartphone vibrations. While smartphones have long been vulnerable to various covert channels, the use of mmWave radar to intercept and decode vibration-based data transmission remains an unexplored area. This research demonstrates how mmWave radar can remotely capture and decode data encoded in vibrations, offering a new perspective on smartphone security risks.

Many existing vibration-based covert channels require internet access, close proximity, or another communication method to transmit data to an attacker. However, this approach enables remote communication, allowing an attacker to retrieve data from a distance without the need for direct access to the target device.

By shedding light on this emerging attack vector, this study emphasizes the need for further investigation into defensive strategies against such covert exfiltration techniques.

Acknowledgment

I am truly grateful to the University of Colombo School of Computing (UCSC) for giving me the platform to explore and carry out this research. The academic environment and support I received have been invaluable in shaping my learning and growth.

A special thanks goes to my research supervisor, Dr. Asanka Sayakkara, for his patience, insightful advice, and constant encouragement. His guidance has been a driving force behind my progress and has helped me overcome many challenges along the way.

I would also like to express my heartfelt appreciation to my family and friends for their endless support, motivation, and understanding. Their belief in me has given me the strength to keep pushing forward, and I am truly grateful for their presence on this journey.

Contents

1	Introduction	1
1.1	Research Problem	2
1.2	Research Questions	2
1.3	Research Aims and Objectives	3
1.4	Significance of the Research	3
1.5	Scope and Delimitations	4
2	Background and Related Work	6
2.1	Background	6
2.1.1	Vibration Motor	6
2.1.2	mmWave sensing	6
2.1.3	Smartphone vibration detection	7
2.2	Related Work	8
2.2.1	Covert Channels on Smartphones	9
2.2.2	Smartphone Vibrations as a Covert Channel	9
2.2.3	Detection of Vibrations using mmWave Radar	10
2.2.4	Summary	10
3	Research Methodology and Design	12
3.1	Research Methodology	12
3.2	Research Design	15
3.2.1	Sender Application	15
3.2.2	Receiver Design	18
4	Implementation	22
4.1	Hardware setup	22
4.1.1	Radar Device and Configuration	23
4.1.2	Smartphone Setup	23
4.1.3	Determining Phone and Radar Positioning	24
4.2	Determining Vibration Amplitude	24
4.3	Determining Vibration Period	26
4.4	Encoding data	27
4.5	Signal Capture Using mmWave Studio	32
4.6	Decoding transmitted data	32
4.6.1	Preprocessing raw radar data	33
4.6.2	Target Detection and Signal Extraction	33
4.6.3	Signal Processing and Demodulation	36

5	Evaluation and Results	40
5.1	Evaluation matrix	40
5.1.1	Bit Error Rate (BER)	40
5.1.2	Data Rate	40
5.1.3	Participant-Based Stealth Evaluation	41
5.2	Vibration Audibility Across Different Surface Materials	41
5.3	Evaluation of Different Encoding Schemes	43
5.3.1	Data frame structure	44
5.3.2	Encoding Data into Vibration Signal	44
5.3.3	Decoding Amplitude Modulated Vibration signal	45
5.3.4	Comparison of On–Off Keying (OOK) and Pulse Width and Amplitude Modulation (PWAM)	47
5.4	Evaluation Based on Distance	48
5.5	Impact of Barriers on Signal Propagation	50
5.6	Evaluation with Different Smartphone Models	52
6	Conclusion and Future Work	52
6.1	Conclusion for Research Question 1	53
6.2	Conclusion for Research Question 2	53
6.3	Conclusion for Research Problem	54
6.4	Limitations and Future Directions	54

List of Figures

1	Smartphone Vibration	7
2	Research Methodology	13
3	Attack Model	15
4	Vibration inertia	17
5	Plots generated by mmWave Studio	18
6	Detection and Angle Estimation Results plot	19
7	Transmitter identification figures	21
8	Hardware Setup	22
9	Audibility of different vibration amplitudes	25
10	Phase variations observed at low vibration amplitudes.	25
11	BER% vs Vibration period	27
12	Frame structure with encoded data	28
13	Vibration frame with On Off keying modulation	30
14	Top: Normalized amplitude graph Bottom: Binary signal generated using thresholding	36
15	Binary signal containing transmitted frame structure	38

16	Vibration audibility perception across different surface materials . . .	41
17	Amplitude vs. time representation of the vibration levels used in PWAM	43
18	Frame structure with Pulse Width Amplitude Modulated data	44
19	Demodulation	45
20	Vibration decodability at different distances	48
21	Phase variation of received signal at increasing distances.	49
22	BER for Different Barrier Materials	51
23	Phase variations observed when different barrier materials were placed between the radar and the smartphone.	51

List of Tables

1	Bit Error Rate (BER) Across Different Phone Orientations	24
2	Bit Error Rate at different amplitude levels	25
3	Participant distribution across different age groups	42
4	Comparison of OOK and PWAM encoding Schemes	48
5	BER for Different Cardboard Thicknesses	50
6	BER Performance across Different Pixel Smartphone Models	52

Abbreviations

mmWave	Millimetre -Wave
FMCW	Frequency-Modulated Continuous Wave
FFT	Fast Fourier Transform
OOK	On–Off Keying
DCA	Data Capture Adapter
EVM	Evaluation Module
BASK	Binary Amplitude Shift Keying
SDK	Software Development Kit
LRA	Linear Resonant Actuators
ERM	Eccentric Rotating Mass

GUI	Graphical User Interface
DFP	Device Firmware Package
ADC	Analog-to-Digital Converter
CFAR-CA	Constant False Alarm Rate – Cell Averaging
CFAR	Constant False Alarm Rate
DC	Direct Current
BER	Bit Error Rate
PWAM	Pulse Width and Amplitude Modulation
IF	Intermediate Frequency

1 Introduction

Millimetre -Wave (mmWave) radars, a form of Frequency-Modulated Continuous Wave (FMCW) radar, are gaining attention due to their high-frequency operation, allowing for greater bandwidth, high-resolution imaging, and precise detection of objects (Radar and Rao (2017)). By emitting a frequency-modulated signal and analyzing its reflection, mmWave radars can determine an object’s range, velocity, and angle, while even detecting minute vibrations in objects via variations in reflected signal frequency and phase (Cui et al. (2023)). This capability opens opportunities for using mmWave radar in novel applications like covert data transmission from smartphones.

Covert communication involves transmitting information that cannot be detected, posing a critical challenge to data security. Smartphones are particularly vulnerable targets as they store sensitive user data and provide multiple exploitable channels for data exfiltration. Attackers have leveraged various covert channels to extract and transmit information without user awareness, including vibration patterns (Al-Haiqi et al. (2014)), ultrasonic and isolated sound channels (Deshotels (2014)), network covert channels (Gasior and Yang (2011)), cellular voice channels (Aloraini et al. (2015)), battery usage and even phone calls (Chandra et al. (2015)). These methods exploit permissions granted to smartphone applications to stealthily extract and transmit sensitive information.

In this research, we will utilize the built-in vibration motor in smartphones to generate vibration patterns for covert data transmission. The research involves encoding data into vibration signals, which will be captured and decoded by mmWave radar. Signal processing techniques, such as Fourier Transform and phase shift detection, will be employed to decode these vibrations and retrieve the transmitted data.

Among these, vibration-based covert channels have been extensively studied. Schlegel et al. (2011) introduced one of the earliest approaches, Soundcomber, It encodes stolen data into vibration mode changes, while a second app detects these changes and transmits the data over the internet to the attacker. Similarly, Al-Haiqi et al. (2014), Casolare et al. (2021), and Novak et al. (2015) studies use a source app to generate vibration patterns and a sink app to decode them via the accelerometer, but they also require internet access for data exfiltration, as attackers cannot remotely sense vibrations. Deshotels (2014) proposed an inter-device vibration channel, but it relies on two phones in physical contact for data transfer. Additionally, most of these methods use maximum vibration intensity, making them detectable by users. This research overcomes these limitations by enabling remote data exfiltration without any direct connectivity to the smartphone and using reduced vibration intensity to remain unnoticed.

This research overcomes these limitations by proposing a novel vibration-based covert communication method using mmWave radar. Unlike previous approaches, this method eliminates the need for any direct connectivity to the smartphone by enabling remote detection of vibrations via mmWave radar. Furthermore, by using reduced vibration intensity, the method remains stealthy and difficult to detect. This study will explore the feasibility of using mmWave radar for covert data exfiltration, focusing on signal processing techniques, data encoding and decoding methods, and the optimal conditions for reliable transmission. By exposing this new attack vector, the research highlights new vulnerabilities in smartphone security, which helps to strengthen defences against unauthorized data exfiltration and emphasises the significance of considering modern technology like mmWave radar in the context of cybersecurity threats.

1.1 Research Problem

Previous studies have explored various channels for covertly transmitting data from smartphones, including through vibrations. However, the novel approach of using mmWave radar to detect and exfiltrate smartphone data covertly through vibrations remains unexplored. While some research has investigated mmWave radar’s ability to sense smartphone vibrations, none have specifically addressed the scenario of covert data transmission, where the vibration signals generated would vary based on the transmitted data and cannot be decoded using predefined pattern templates.

This research aims to fill these gaps by integrating covert data transmission from smartphones with the sensing capabilities of mmWave radar technology. By doing so, it aims to demonstrate a new application of mmWave technology in the field of cyber security.

1.2 Research Questions

1. To what extent does a mmWave FMCW radar can detect vibrations of a smartphone?

Hypothesis: Reducing the transmitted amplitude and changing the vibration period will affect both the detection range and the accuracy of data received by the radar. This research hypothesizes that employing effective encoding patterns and advanced signal processing algorithms will improve the extent to which radar can detect and interpret smartphone vibrations accurately.

2. What methodology can be used to effectively transfer data from a smartphone covertly into a mmWave FMCW radar using vibration patterns?

Hypothesis: Reducing the transmission amplitude and sending data only

when the smartphone is not actively in use will enhance the covert nature of data transfer.

1.3 Research Aims and Objectives

Aim:

The aim of this research is to demonstrate the feasibility of covert data transmission between smartphones and mmWave FMCW radar using vibration patterns, highlighting potential vulnerabilities in smartphone security.

Objectives:

- Explore the utilization of smartphone vibrations and mmWave radar technology for covert data transmission.
- Develop a methodology for encoding sensitive data into vibration signals generated by the smartphone's vibration motor.
- Investigate the capabilities of mmWave radar in capturing and decoding smartphone-generated vibration signals.
- Evaluate the reliability and effectiveness of the covert communication channel under various real-world scenarios, including different smartphone models, distances between the smartphone and mmWave radar, and environmental conditions.
- Identify potential vulnerabilities in smartphone security systems exposed by this covert data exfiltration method.

1.4 Significance of the Research

In terms of research contribution to computer science, it introduces a novel approach for sensing covert data transmitted by smartphone vibrations using mmWave radar technology. Combining these two areas advances our understanding of covert communication techniques and extends the application of mmWave radar in cyber security by providing new insights into potential vulnerabilities in smartphone systems, giving us more ways to defend against potential attacks and, conversely, more tactics for attackers to try in the digital world.

Furthermore, the project highlights the need to address security vulnerabilities inherent in smartphone systems. Unlike other covert data exfiltration methods that may require internet access or close proximity to the target device, the utilization of mmWave radar presents a greater threat. With its ability to detect minute vibrations

emitted by smartphones from a distance, attackers could potentially access sensitive data without the need for explicit permissions or physical proximity to the device.

By highlighting these vulnerabilities and exploring novel detection methods, the project aims to foster greater awareness and resilience against emerging threats in the digital landscape.

1.5 Scope and Delimitations

In Scope

- Identification of covert communication method
Test different approaches to covertly communicate from the smartphone to send data through vibrations.
- Data encoding and transmission
Encode data into vibration signals and transmit it from the smartphone in a manner that allows a large amount of data to be sent quickly and remains undetectable.
- Vibration signal decoding
Identify smartphone vibrations by isolating them from other signals. Perform error correction, decode the vibration signals, and retrieve the data.
- Orientation and distance
Evaluate and determine the optimal orientation and distance that support covert data exfiltration. The setup must ensure that the communication remains undetectable by the user.
- Compatibility of different smartphones
Test the method with different smartphones to ensure accuracy and reliability.

Out Scope

- Multiple object detection

The research focuses exclusively on monitoring vibration signals from a single smartphone at a time. Communication between multiple smartphones and the mmWave radar is not addressed in this study.

- Smartphone In Motion

This research assumes that the smartphone is stationary. Scenarios where the smartphone is in motion, such as when held by the user or carried in a pocket, are not considered feasible for covert communication via vibrations and are thus not included in this study.

- Functional application development

The mobile application developed for this research will be minimal, designed solely to retrieve sensitive data and transmit it via vibrations. While the app will appear to be a legitimate application with the necessary permissions, this research focuses only on the data transmission aspect and does not involve developing the full, user-facing functionality of the app.

- Alternative Data Transmission Methods

Methods such as optical, acoustic (non-vibration based), or other unconventional data transmission techniques are excluded.

- Secured Data transmission

The study does not involve securing transmitted data with encryption algorithms or cryptographic protocols.

- Evaluated only with Android devices

This study exclusively focuses on evaluating the radar's ability to detect and interpret smartphone vibrations using Android devices. Therefore, it does not include an assessment of other operating systems, such as iOS.

- Limited to text data transmission

This research is confined to the transmission of text-based data through smartphone vibrations, excluding more complex data types such as images, audio, or video.

2 Background and Related Work

2.1 Background

2.1.1 Vibration Motor

A vibration motor generates vibrations by moving a metallic mass periodically. The two main types are Eccentric Rotating Mass (ERM), which uses a rotating eccentric mass, and Linear Resonant Actuators (LRA), which move a magnetic mass linearly (Roy et al., 2015). LRAs are used in most mobile phones due to their fast response time. It can vibrate in different directions: Z-axis LRAs oscillate perpendicularly to the screen, while X-axis LRAs vibrate laterally. This allows independent control of vibration frequency and amplitude, making them ideal for precise haptic feedback (Cui et al., 2023).

2.1.2 mmWave sensing

Millimeter-wave (mmWave) radar transmits FMCW chirp signals to detect objects. A time-delayed version of the transmitted signal is reflected back from the object. The transmitted and received signals can be expressed as follows:

$$\text{TX}(t) = \exp [j (2\pi f_c t + \pi S t^2)] \quad (1)$$

$$\text{Rx}(t) = \text{TX}(t - t_d) = \exp (j 2\pi f_c (t - t_d) + j \pi S (t - t_d)^2) \quad (2)$$

where $t_d = \frac{2d}{c}$ represents the time delay, f_c is the starting frequency of the chirp, and S denotes the chirp slope.

The mmWave radar mixes the transmitted and received signals, producing an Intermediate Frequency (IF) signal, $s(t)$, which consists of a tone with a beat frequency f_{IF} :

$$\begin{aligned} s(t) &= \text{Rx}^*(t) \text{Tx}(t) \\ &= \alpha \exp [-j (2\pi S t_d t + 2\pi f_c t_d - \pi S t_d^2)] \\ &\approx \alpha \exp \left[-j \left(\underbrace{2\pi \frac{S \cdot 2d}{c} t}_{f_{IF}} + \underbrace{\frac{4\pi d}{\lambda}}_{\phi} \right) \right] \end{aligned} \quad (3)$$

where λ is the wavelength of the transmitted signal. Since the beat frequency f_{IF} contains distance information, the distance d between the object and the radar can be estimated by performing a Fast Fourier Transform (FFT) on the IF signal.

To detect very small movements, such as micro-displacements, the radar relies on phase changes rather than frequency shifts. While range resolution is determined by the chirp bandwidth, movements smaller than this resolution do not significantly shift the beat frequency. Instead, these small displacements induce phase variations in the IF signal. By tracking these phase variations, the radar can detect minute changes in distance.

From Eq 3, the phase of the IF signal, $\phi(t)$, is given by:

$$\phi(t) = \frac{4\pi d}{\lambda}. \quad (4)$$

If the object's distance changes by Δd , the phase value will change accordingly. Thus, the displacement Δd can be derived from the phase shift $\Delta\phi$ using the following relationship (Jiang et al., 2020):

$$\Delta d = \frac{\lambda \Delta\phi}{4\pi}. \quad (5)$$

2.1.3 Smartphone vibration detection

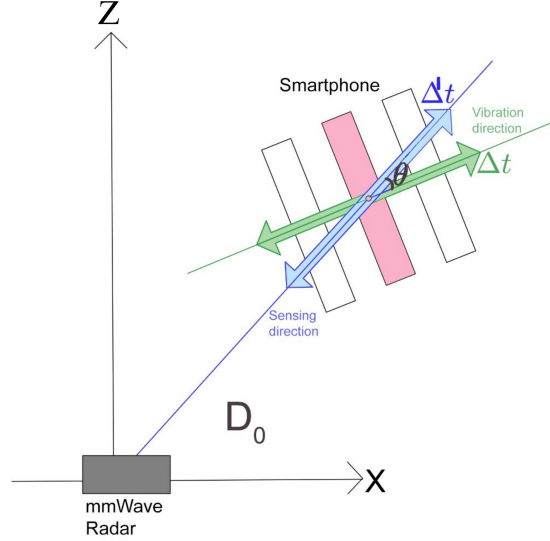


Figure 1: Smartphone Vibration

mmWave radar captures smartphone vibrations. Fig. 1 presents a basic vibration model originally proposed by Cui et al. (2023). As the smartphone vibrates, the radar senses the motion, which follows a simple harmonic pattern and undergoes a time-dependent micro-displacement $\Delta(t)$, which is given by:

$$\Delta(t) = A \cos(2\pi f_v t) \quad (6)$$

where A represents the vibration amplitude, and f_v is the vibration frequency. Due to the angular misalignment θ between the smartphone's vibration direction and the sensing direction of the mmWave radar, the effective displacement detected by the radar, denoted as $\Delta'(t)$, is the projection of $\Delta(t)$ along the sensing axis:

$$\Delta'(t) = \cos \theta \cdot \Delta(t) \quad (7)$$

Thus, the instantaneous distance $d(t)$ measured by the radar is given by:

$$d(t) = d_0 + \Delta'(t) \quad (8)$$

By substituting $d(t)$ for the object distance d in Eq. 3, the received Intermediate Frequency (IF) signal $r(t)$ from the smartphone can be expressed as:

$$r(t) = \alpha \exp \left[-j \left(\frac{4\pi S}{c} d(t)t + \frac{4\pi}{\lambda} d(t) \right) \right] \quad (9)$$

To isolate the reflected signal $R_s(t)$ from the target's range, we apply a Range-FFT to the IF signal:

$$r(t) \xrightarrow{\text{Range-FFT}} R_s(t) = \alpha \exp \left[-j \frac{4\pi}{\lambda} d(t) \right] \quad (10)$$

The corresponding phase measurements for the detected target can then be written as:

$$\Phi_s(t) = \frac{4\pi}{\lambda} (d_0 + \cos \theta \cdot \Delta(t)) \mod 2\pi \quad (11)$$

Thus, the extracted phase signal $\Phi_s(t)$ effectively represents the smartphone's vibration pattern as detected by the mmWave radar (Cui et al., 2023)

2.2 Related Work

Studies on covert channel data exfiltration have been conducted for many years and on many emerging technologies in different ways. This review aims to explore how vibrations can be used for covert communication on smartphones and how mmWave radar can be employed to detect and decode these vibrations.

2.2.1 Covert Channels on Smartphones

With the increase in the number of users using smartphones, the amount of personal and sensitive information they hold is also immense. This has led to the development of new attack strategies to steal information from smartphones. Many studies have been conducted on covert data exfiltration from smartphones, utilizing various channels available on these devices to transmit stolen data to attackers in different ways.

Al-Haiqi et al. (2014) describes methods to encode data into vibration patterns and use vibration motors in smartphones to generate vibrations, which can be captured by the accelerometer in the same smartphone to decode the information. This technique is primarily designed for communicating sensitive information from one app to another within the same mobile device. Deshotels (2014) presents two sound-based covert channels: ultrasonic sound and isolated sound. Ultrasonic sounds leverage the smartphone’s capability to emit sounds beyond the human audible range (above 18 kHz), which can be captured by a microphone within the phone or another smartphone. Isolated sounds in this research are similar to Al-Haiqi et al. (2014) as they also use vibration motors to produce short vibrations that are not loud enough to be heard, capturing those vibrations with an accelerometer to decode the data.

Gasior and Yang (2011) presents a method for using network covert channels to leak data from Android devices. Additionally, cellular voice channels can also be used to transmit data covertly, as described in Aloraini et al. (2015). This method uses an audio modem and rootkit to open the cellular voice channel for data transmission during incoming and outgoing voice calls, transmitting data as audio tones covertly to the attacker. Chandra et al. (2015) provides a systemic analysis of shared resources on mobile phones that can support covert communication and introduces two new covert communication channels: battery usage and phone calls.

2.2.2 Smartphone Vibrations as a Covert Channel

The previous section explored various covert channels available in smartphones, and this research focuses on using vibrations as a covert communication channel due to the ability of mmWave radars to effectively detect and decode vibrations.

In Al-Haiqi et al. (2014), Casolare et al. (2021), and Deshotels (2014), an app with access to sensitive information and vibration permissions translates each character using ASCII codes, representing them with 7 bits. These bits are then encoded into vibration patterns for data exchange. The vibration patterns are extracted from accelerometer data to retrieve the stolen information. These studies also take into account factors like whether the user is holding the phone and checking if it’s during typical phone usage periods or nighttime, to determine if the phone is unattended for covert data transmission. Novak et al. (2015) presents another approach that uses

the vibration motor as the sender and the accelerometer as the receiver, but in this case, vibration signals are modulated using Binary Amplitude Shift Keying (BASK).

Roy et al. (2015) describes a method for two smartphones to communicate securely with each other through vibrations. Although this study is not focused on covert channel transmission, it provides insights into achieving higher bit rates during transmission, up to 80 bits/s in smartphones, using techniques such as multi-carrier modulation, orthogonal vibration, and vibration braking. Additionally, the study discusses techniques to ensure secure communication through side-channel jamming.

2.2.3 Detection of Vibrations using mmWave Radar

Basak and Gowda (2022) presents mmSpy, a mmWave based eavesdropping technique that targets smartphone earpiece vibrations. This system operates remotely without any physical contact with the target. The core idea is that millimetre waves reflected from the rear side of a smartphone are highly correlated with the vibrations from the earpiece. By eliminating unwanted noise from the reflected signal, it is possible to recover the actual speech. The reflected signal undergoes statistical error correction preprocessing, signal filtering, and speech recognition using a convolutional neural network-based model. Cui et al. (2024) utilizes smartphone speakers to send messages to the millimeter wave radar. Vibrations generated by the speaker can be detected by the mmWave radar to decode the message, achieving a higher data rate compared to other vibration-based methods. Similarly, Cui et al. (2023) can send requests to mmWave radars by modulating vibration motors. Walsh codes are used to construct vibration patterns to represent different user requests. These Vibration signals can be extracted with high accuracy using predefined vibration patterns and frequencies, but this approach has a limited data rate due to the narrow bandwidth of vibration motors. The main motive of this research is to send short messages to mmWave radar, such as pedestrian crossing alerts, where a lower data rate is sufficient.

2.2.4 Summary

The reviewed literature highlights diverse covert channels for data exfiltration on smartphones, leveraging methods such as vibration, sound, and network channels. Notably, several studies focus on encoding data within vibration patterns, which can then be captured and decoded via accelerometers within the device (e.g., Al-Haiqi et al. (2014), Deshotels (2014)). These methods allow covert communication by using vibration motors to transmit sensitive data, though their application is generally restricted to within-device or close-range transmission. In other studies, sound-based and network-based covert channels expand this idea, demonstrating the adaptability of smartphones for stealthy data transmission.

More recent research has explored mmWave radar’s ability to detect smartphone vibrations and decode data, as shown in Basak and Gowda (2022) and Cui et al. (2023). While these studies successfully utilize mmWave radar to recover data from device vibrations, they are primarily designed to transmit predefined messages and rely on fixed vibration templates. This approach results in limited flexibility, especially in applications requiring dynamic, variable data transmission rates. Moreover, these studies do not address covert data transmission, as their focus is on intentional and structured communication rather than undetectable or stealthy data exfiltration.

This research seeks to bridge the gap by applying mmWave radar technology to covertly detect and decode smartphone data from vibration signals that are not restricted to predefined templates, enabling more versatile covert communication. This novel approach expands the potential applications of mmWave radar in cybersecurity, as it demonstrates an adaptable method for data exfiltration that could evade existing detection mechanisms.

3 Research Methodology and Design

This chapter outlines the overall methodology adopted to achieve the research objectives, encompassing both the practical implementation and the underlying design decisions. It details the end-to-end process of developing the covert communication channel using mmWave radar and smartphone vibrations, as well as the criteria used to evaluate its performance.

3.1 Research Methodology

The Figure 2 provides a high-level overview of the key stages involved in the study, from system setup to performance evaluation.

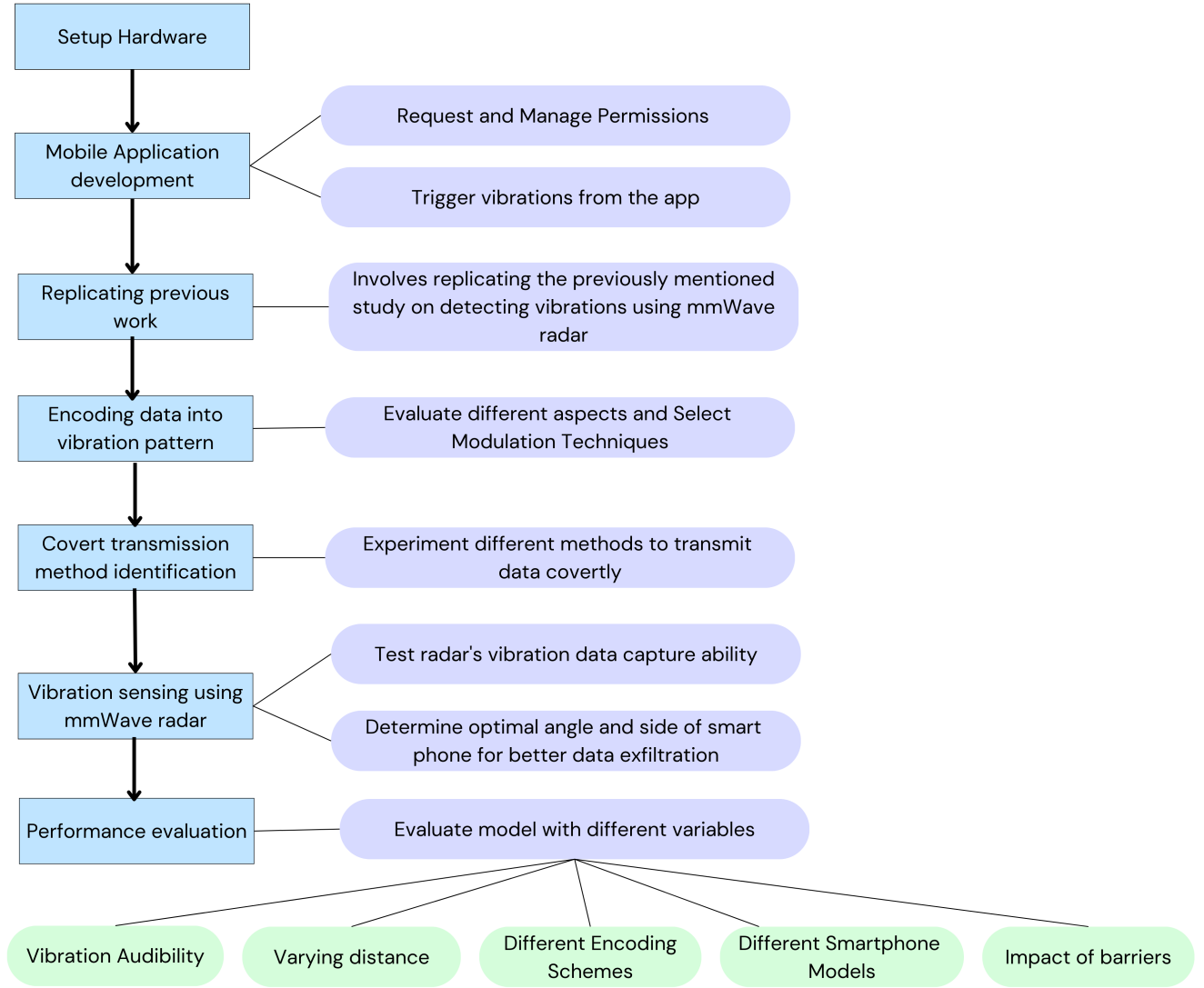


Figure 2: Research Methodology

Each component of the methodology depicted in Figure 2 is described in detail below:

1. Hardware and software setup

In this initial step, all required equipment, including the mmWave radar, Data Capture Adapter (DCA)1000 Evaluation Module (EVM), and smartphones, are set up and calibrated to ensure accurate vibration detection and analy-

sis. Additionally, essential software like mmWave Studio, UniFlash, and the mmWave Software Development Kit (SDK) must be installed to configure the radar and process the captured signals.

2. Mobile Application Development

The mobile app is designed to control the smartphone’s vibration motor, with the necessary permissions for triggering vibrations. This app will trigger vibrations with specific patterns and intensities that can encode data. The app effectively becomes the main tool for translating data into vibrations that the radar can detect.

3. Replicating Previous Work

This step involves replicating earlier research to establish a benchmark, confirming that mmWave radar can indeed detect smartphone vibrations under controlled conditions. By reproducing key findings from studies such as mm-Ripple Cui et al. (2023), we validate our hardware setup and baseline results, accounting for any variations due to equipment differences or environmental factors. This step verifies that the system works as expected before moving on to the novel aspects of the study.

4. Encoding Data into Vibration Pattern

In this step, we concentrate on developing a method to encode data within specific vibration patterns and identifying the most effective encoding technique. Various modulation methods are also tested to determine which provides the most reliable data transmission. These methods are evaluated for signal clarity, aiming to maximize data accuracy while minimizing errors.

5. Covert Transmission Method Identification

To keep data transmission covert, we explore various techniques for sending information through vibrations in a way that remains undetectable. This includes testing different vibration patterns and intensities to avoid detection while preserving data integrity and triggering vibrations only when the phone is idle. These techniques are refined to make transmissions as discreet as possible.

6. Vibration Sensing Using mmWave Radar

The radar’s effectiveness in detecting the detailed vibration patterns created by the smartphone is tested in this step. Optimal positioning is identified to maximize signal clarity, determining the best angle and side of the phone to improve data capture. Consistency across various positions and orientations is tested to standardize setup guidelines for reliable readings. This step ensures that the radar can accurately capture encoded vibrations.

7. Performance Evaluation

Finally, the developed model is evaluated under various conditions to determine its robustness. Testing includes evaluating the approach with different barriers, distances, encoding techniques, smartphone models and testing the perceptibility of vibration covert channel to understand how each variable impacts the proposed communication channel.

3.2 Research Design

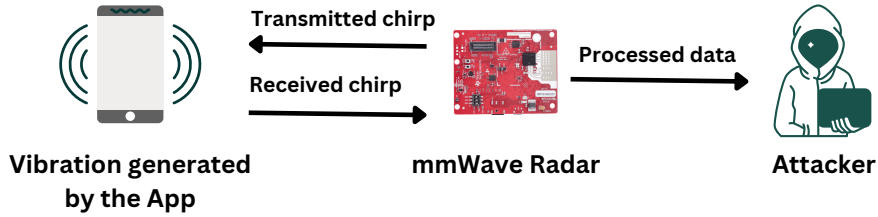


Figure 3: Attack Model

As shown in Figure 3, this research presents a general model that consists of a mobile application running on the target user’s device and a mmWave radar positioned remotely to capture covertly transmitted data. The mobile application encodes sensitive data into vibration patterns, which are then detected and decoded by the radar. The following subsections describe each component in detail.

3.2.1 Sender Application

The sender application requires two key permissions:

1. Access to sensitive data (e.g., contacts, messages, microphone).
2. Permission to generate vibration patterns using the device’s vibration motor.

Both permissions can be granted through legitimate-seeming functionality, making the application appear harmless. One factor that may make the application appear less suspicious is not requiring network access permissions, giving the impression that no data is being transmitted externally.

The following Android Permissions can be used to enable covert transmission without raising suspicion:

- **Vibrator Class Permission:** The Vibrator class allows an application to control the device’s vibration motor by carefully modulating vibration intensity and duration. Data can be encoded into vibration signals that are undetectable to the user but detectable by a mmWave radar.

- **Partial Wake Lock Permission:** The `PowerManager.PARTIAL_WAKE_LOCK` keeps the CPU active while allowing the screen to turn off. This ensures that the application can continue generating vibrations even when the phone appears to be idle Al-Haiqi et al. (2014).
- **Accelerometer Access Permission:** The `onSensorChanged()` method from the `SensorEventListener` class allows the application to monitor device motion. If the phone remains stable for an extended period, it is likely resting on a surface, making it an ideal time for covert data transmission Casolare et al. (2021).

These permissions do not require explicit user approval, allowing the application to perform its covert functions in the background without raising suspicion. However, to access sensitive user data (such as messages, audio, or contacts), the application must request additional permissions that require user approval.

For example, a malicious application disguised as a note-taking app can request the `RECORD_AUDIO` permission, claiming to support voice notes. Once granted, the app can eavesdrop on conversations and covertly encode the extracted data into vibration signals. Since the app does not require internet access, this will not raise any suspicion on data transmission.

To enable covert data transmission, the malicious application must be programmed to send data at a predetermined time set by the attacker. This timing is crucial, as the attacker must be within a specific range (up to 3.1 meters) to capture the transmitted data using mmWave radar. Before initiating the transmission, the app monitors the device’s motion state using the accelerometer sensor. By continuously analyzing sensor readings through the `onSensorChanged()` method, the app can determine if the phone has been stationary for an extended period, indicating that it is idle and resting on a surface. Once the phone is confirmed to be idle, the app uses a Partial Wake Lock to keep the CPU active while allowing the screen to remain off. This ensures that vibrations can be generated even when the phone appears inactive. The application then employs the `Vibrator` API to create low-intensity, nearly imperceptible vibrations, encoding data within specific vibration patterns. These subtle vibrations are undetectable to the user but can be remotely detected and decoded by an attacker using mmWave radar, effectively enabling data exfiltration without requiring direct access to the device.

Vibration Pattern Format

Sensitive data is transmitted using vibration patterns. Each character is encoded as an 8-bit binary ASCII code and then encoded with Hamming (7,4) for error correction, resulting in a 14-bit representation per character. A structured frame format is used for data transmission, as described below:

- **Preamble:** The frame begins with a preamble, represented by 0xAA (8 bits).

- **Data Length:** The next 8 bits indicate the length of the data being transmitted. The maximum representable data length with 8 bits is 255. If the data exceeds 255 bits, the remaining data is sent in subsequent frames.
- **Data Segment:** Contains the actual encoded information and has a length specified by the data length field.

Vibration Signal Generation

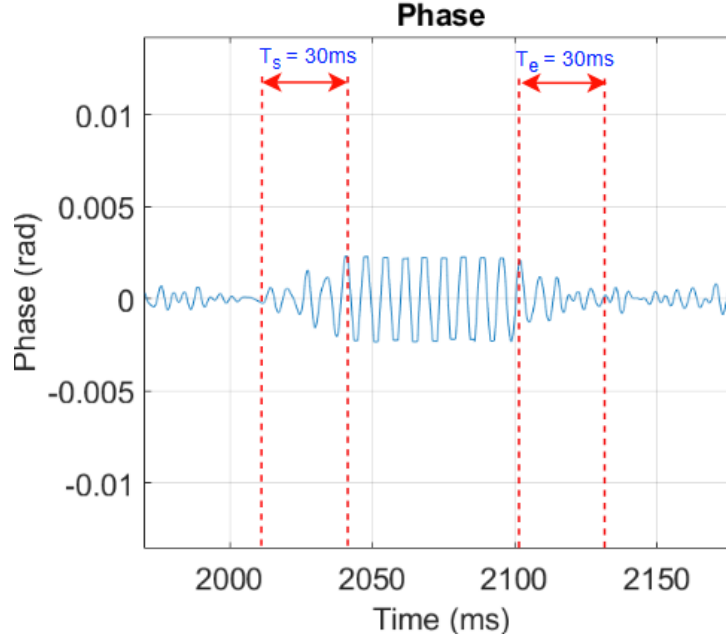


Figure 4: Vibration inertia

The Vibrator class allows control over smartphone vibrations using time and amplitude settings, where amplitude can range from 0 to 255.

One important factor is that the vibration motor does not reach its full strength instantly when turned on, nor does it stop immediately when turned off. This is due to the motor’s inertia, which causes a delay in both starting and stopping. These delays are known as the rise time and fall time of the motor, respectively.

To ensure reliable data transmission, the vibration duration must be long enough for the motor to reach a stable state. If it’s too short, the motor may not fully vibrate before stopping, making it harder for the radar to detect and decode the signal correctly. Prior research in Cui et al. (2023) has shown that the rise time can vary between 4 ms to 120 ms, depending on device and amplitude. In our tests with the Google Pixel 3 placed 0.64 meters away, we observed that the rise time is

approximately 30 ms. As shown in Figure 4, the vibration signal gradually increases, for a period of T_s and fades out during T_e , which marks the fall time.

We empirically tested various time slot durations and found that a minimum period of 80 ms provides sufficient time for the motor to reach full strength and stop cleanly, resulting in the lowest BER. Therefore, we selected 80 ms as the optimal vibration time slot duration for all transmissions.

Additionally, the vibration amplitude must remain low enough to avoid perceptibility. We evaluated several amplitude levels for both audibility and radar decodability, ultimately selecting 20% of the maximum value (i.e., amplitude = 50) as the optimal trade-off between stealth and signal strength.

3.2.2 Receiver Design

mmWave Studio Post-Processing and Target Detection

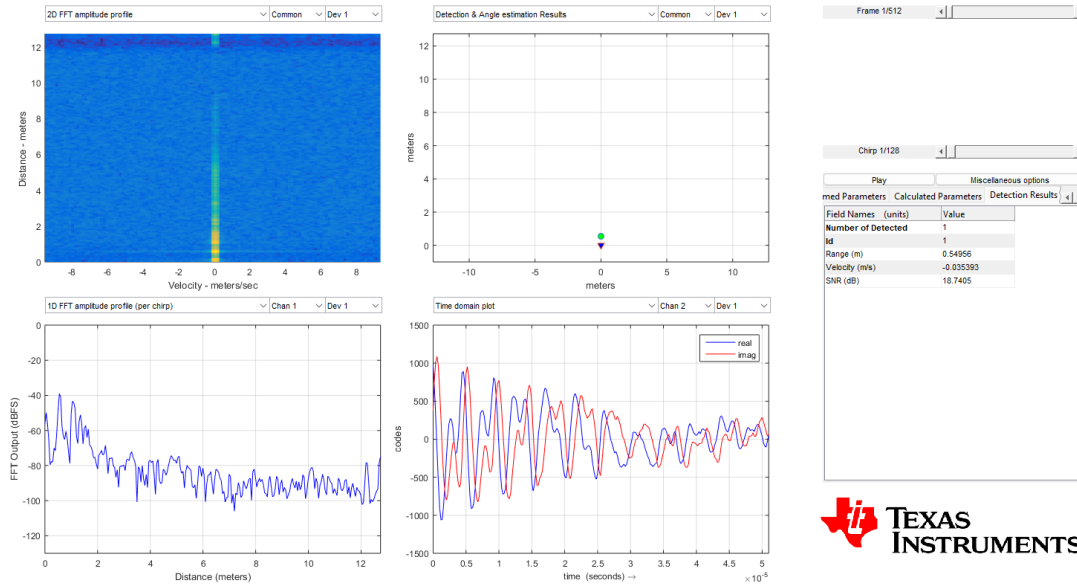


Figure 5: Plots generated by mmWave Studio

The mmWave Studio software is used to configure the radar device and collect Analog-to-Digital Converter (ADC) data for offline processing. This also has a post-processing tool (PostProc) which is used to visualize the ADC data collected on a PC. It operates using two key inputs: the raw data captured (stored as a .bin file) and the sequence of APIs used to program the radar device (stored as a .log file). By utilizing these sources, PostProc interprets the collected radar data and generates meaningful plots for analysis. Importantly, mmWave Studio automatically provides these dump

files and API sequences to the MATLAB-based PostProc tool, eliminating the need for manual input.

The PostProc tool offers four different independently selectable plots as shown in Figure 5, each plot is configurable for different channels and data visualization types. Among these, the Detection and Angle Estimation Results plot is critical for this research.

CFAR-Based Detection Graph

The Detection and Angle Estimation Results plot shown in Figure 6 in mmWave Stu-

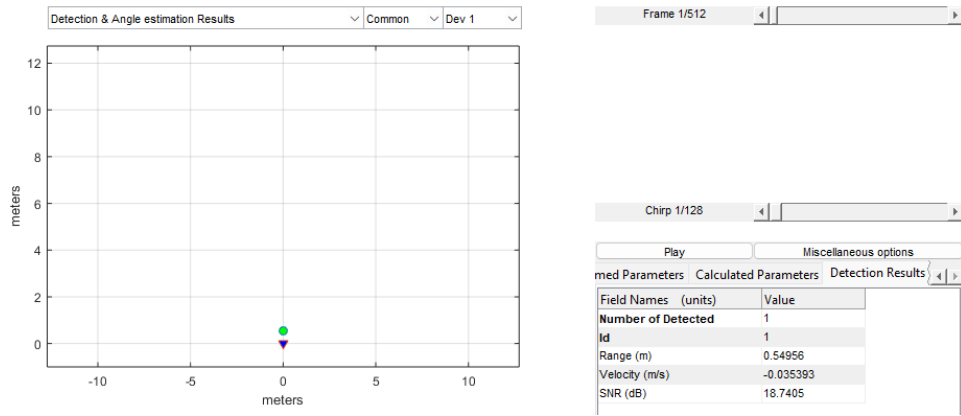


Figure 6: Detection and Angle Estimation Results plot

dio is generated using a 2D-Constant False Alarm Rate – Cell Averaging (CFAR-CA) algorithm, which detects targets within the surrounding by dynamically adjusting detection thresholds. The parameters of the Constant False Alarm Rate (CFAR) algorithm can be fine-tuned in the "Miscellaneous Options" menu to optimize detection accuracy. In the generated plot:

- Red dots indicate objects with negative velocity.
- Green dots indicate zero velocity.
- Blue dots indicate objects with positive velocity.

The radar system assumes an antenna configuration with four receiving (Rx) antennas, separated by $\lambda/2$ in a linear array. To improve angle estimation accuracy, the system performs a 3D-FFT followed by a peak search, allowing it to detect a single target per range-velocity bin. This enables the receiver to determine both the distance and angular position of a target object, such as a smartphone transmitting

vibration-based covert data. By leveraging mmWave Studio’s post-processing capabilities, we can accurately locate the transmitting device and extract the range of the target from the ”Detection results” section provided by mmWave Studio. (Texas Instruments, 2022).

Target Signal Extraction Using Range FFT

To extract the target signal, a range-FFT is performed on the raw signal data in our separate processing code. The CFAR-based detection results from mmWave Studio provide the estimated range of the target. This range is fed into our processing code as input and used to determine the corresponding range bin of the object in our code. Since the range bins are spaced based on the radar’s distance resolution, the bin closest to the detected range is selected. Once the correct range bin is identified, the ADC data corresponding to this bin is extracted for further analysis. This extracted signal is then used for additional processing. The integration of mmWave Studio’s CFAR-based detection results with our FFT-based signal processing ensures accurate target identification and precise signal extraction.

Transmitter Identification

After extracting the target signal, it is essential to determine whether the reflected signal originates from the actual transmitter with defined vibration patterns or from a static object in the environment. As shown in Figure 7a and 7b, in the frequency domain, both static and vibrating objects are affected by the Direct Current (DC) component and low-frequency noise. However, unlike static objects, the frequency spectrum of a vibrating object in Figure 7a exhibits a distinct peak within the smart-phone vibration frequency range of 100 Hz to 300 Hz, as identified by Cui et al. (2023). This characteristic allows us to set a threshold to identify vibrating objects (transmitters) based on their unique frequency signatures.

Once the vibrating target is identified, a bandpass filter (BPF) can be applied to isolate the expected vibration signal while suppressing unwanted noise. Based on empirical characterization, the lower and upper cutoff frequencies of the filter are set at 100 Hz and 300 Hz, respectively, covering the vibration frequencies of most smartphones. After filtering, the extracted vibration signal from the reflection data can be analyzed. The phase changes of recovered signals shown in Figure 7c clearly reveal the transmitted vibration patterns when compared to Figure 7d, which was captured with a static phone, confirming the presence of data transmission.

Demodulation and decoding

At the attacker’s end, demodulation and decoding are performed to retrieve the covertly transmitted data. Once the reflected signal from the target device is extracted, the system identifies the beginning of the transmission by detecting a prede-

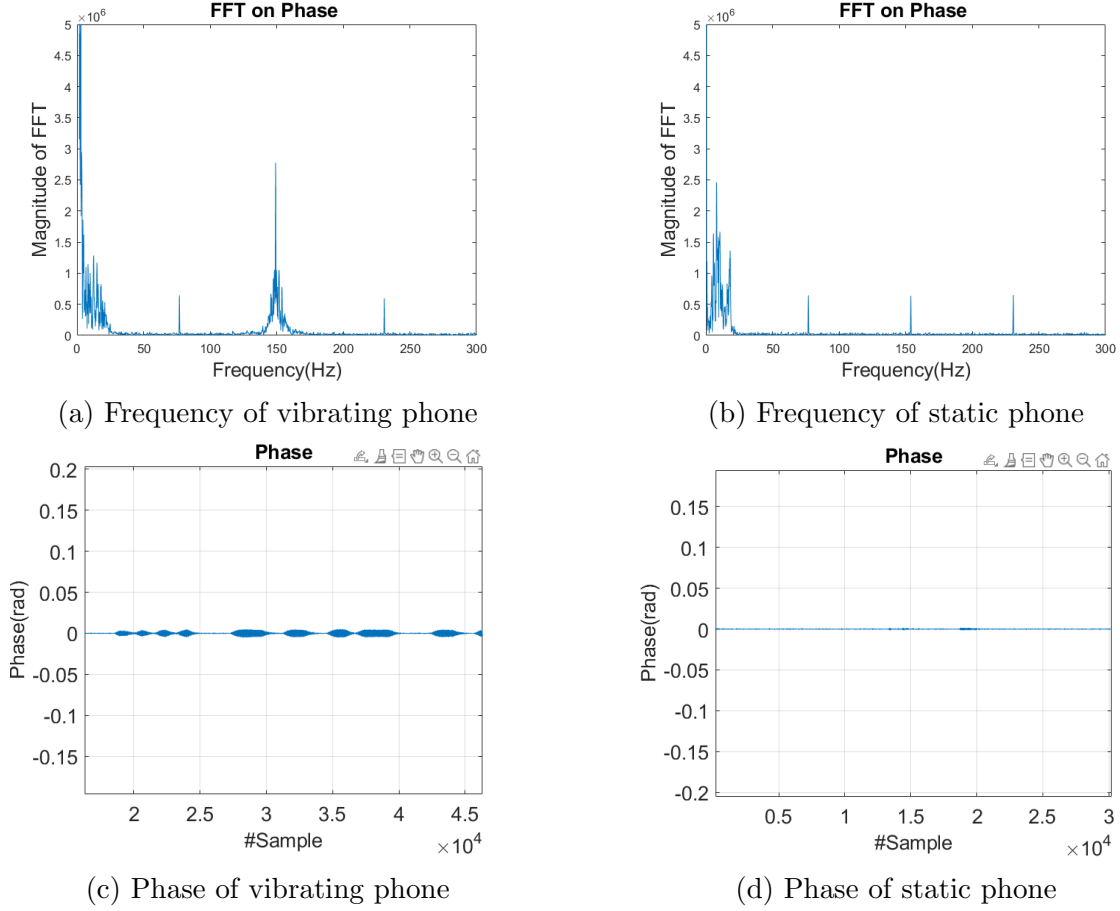


Figure 7: Transmitter identification figures

finer preamble pattern. This preamble serves as a synchronization marker, ensuring accurate data extraction.

Following the preamble, the next 8 bits represent the data length, which is extracted to determine how many subsequent bits should be collected. The system then samples the signal accordingly, using an OOK detection scheme. In this process, each sampled bit is compared against a predefined threshold: if the signal amplitude exceeds the threshold, it is classified as a 1; otherwise, it is classified as a 0.

The decoded bit sequence contains Hamming (7,4) encoded data, which is then processed for error correction. The corrected bits are subsequently converted into their corresponding ASCII characters, reconstructing the original message that was covertly transmitted via smartphone vibrations.

4 Implementation

This section details the practical execution of the proposed system, including the hardware setup, software implementation, and data processing along with important code segments.

4.1 Hardware setup

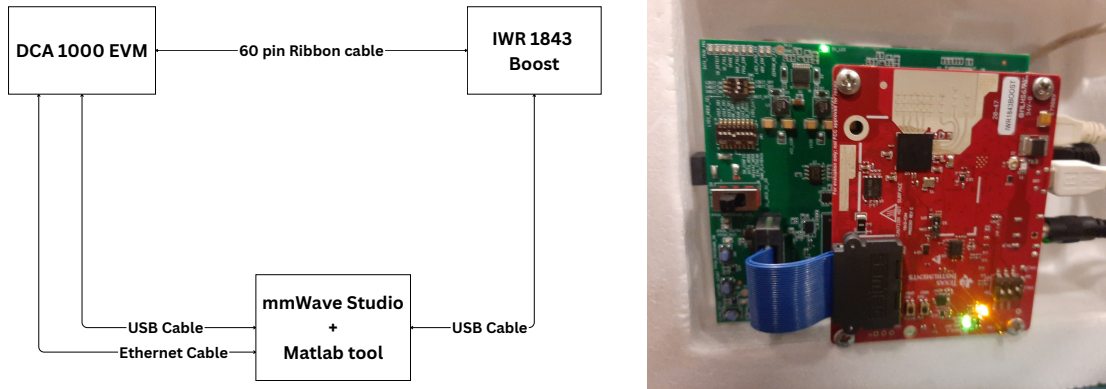


Figure 8: Hardware Setup

The hardware setup for this research is designed as shown in Figure 8 to capture, transfer, and analyze data from a mmWave radar system. This setup includes the IWR 1843 Boost radar sensor, the DCA 1000 EVM data capture card, and a computer running mmWave Studio for configuration and MATLAB for data processing. Functionality of each of these components is explained in detail below:

1. IWR 1843 Boost: This is a radar sensor module that transmits and receives mmWave signals. It's the primary device capturing environmental data through radar waves. The data from this module is transferred to the DCA1000 EVM (Radar and Rao, 2017).
2. DCA1000 EVM: Data Capture Adapter (DCA)1000 Evaluation Module (EVM) is responsible for capturing raw data from the IWR 1843 Boost sensor and sending it to the computer. This data is often in a raw binary (e.g: .bin) format (Instruments, 2019).
3. Computer with mmWave Studio and MATLAB:
 - mmWave Studio: This software is used to configure and control the radar parameters for data capture. mmWave Studio communicates with the radar and the DCA 1000 EVM, coordinating the data acquisition process.

- MATLAB Tool: Once the data is captured in binary format by the DCA 1000 EVM, it can be processed in MATLAB.

4.1.1 Radar Device and Configuration

We configure the radar with a start frequency of 77 GHz, which provides a short wavelength ideal for detecting very small displacements. The ramp time is set to $65 \mu\text{s}$, and with a frequency slope of $58.611 \text{ MHz}/\mu\text{s}$, this results in a total chirp bandwidth of 3809.78 MHz. This bandwidth gives a range resolution of approximately 3.94 cm, calculated using the formula:

$$\Delta R = \frac{c}{2B}$$

where c is the speed of light, and B is the bandwidth. This fine resolution allows the radar to detect small changes in distance. We set the chirp periodicity to 13 ms, ensuring the radar samples are frequent enough to track the periodic motion of the smartphone's vibration motor. To capture multiple vibration cycles, the radar performs 128 chirps per frame and collects 512 frames in total. These settings enable the radar to track phase changes over time, effectively measuring micro-vibrations.

4.1.2 Smartphone Setup

Device Specifications

- Model Used: Google Pixel 3
- Max Amplitude of Vibration: 255 (8-bit)
- Rise/Fall Time of vibration motor: $\approx 30 \text{ ms}$
- Vibration frequency range: 100 Hz - 300 Hz

The smartphone transmits information using OOK, where vibrations are either present (binary 1) or absent (binary 0). To ensure detectability, the vibration intensity was set to the minimum detectable level by the mmWave radar. The transmission was carefully controlled to maintain a stationary phone position, preventing unintended motion artifacts that could interfere with signal interpretation. The radar system captures phase variations induced by these vibrations, enabling robust detection and decoding of the transmitted data.

4.1.3 Determining Phone and Radar Positioning

To determine the correct positioning for both the smartphone and the radar, we conducted a set of experiments evaluating the impact of different phone orientations and placements on the system’s performance. In practical scenarios, it is likely that the radar could be positioned in various orientations relative to the phone, and the phone may also have a back cover. To ensure robustness, we tested the system with the phone facing front, back, sideways, lying flat on a table, and with and without a phone cover, all at a fixed distance of 0.64 meters from the radar.

In the case where the phone was placed flat on a table, the radar was intentionally tilted and positioned such that its antenna directly faced the phone. This configuration was chosen to simulate a realistic yet optimized setup where the radar line-of-sight aligns with the vibration source. This test was particularly important, as the contact surface might distort or absorb the vibration signals. However, the results showed that in all these conditions, the BER remained at 0%, as shown in Table 1. This confirms that the system can successfully detect and decode vibration signals regardless of orientation or surface contact.

Since all orientations produced reliable results with no BER, we adopted the front-facing orientation as the standard setup for the rest of the evaluations. In a real-world covert communication scenario, it is also reasonable to assume that the radar can be positioned in a way that directly faces the front of the target smartphone, aligning with the configuration used in our experiments. In addition to that, Prior results from Cui et al. (2023) indicate that vibration intensity is stronger when the smartphone is placed on a table compared to a tripod. Therefore it can be assumed that if the radar can detect vibrations in the phone holder setup, it will work on a table as well since that produces a greater vibration intensity than when kept on a phone holder.

Table 1: BER Across Different Phone Orientations

Orientation / Condition	BER (%)
Front-facing (on holder)	0.00
Back-facing (on holder)	0.00
Side-facing (on holder)	0.00
Flat on table	0.00
Back-facing with cover	0.00
Back-facing without cover	0.00

4.2 Determining Vibration Amplitude

The vibration amplitude was chosen based on two key factors:

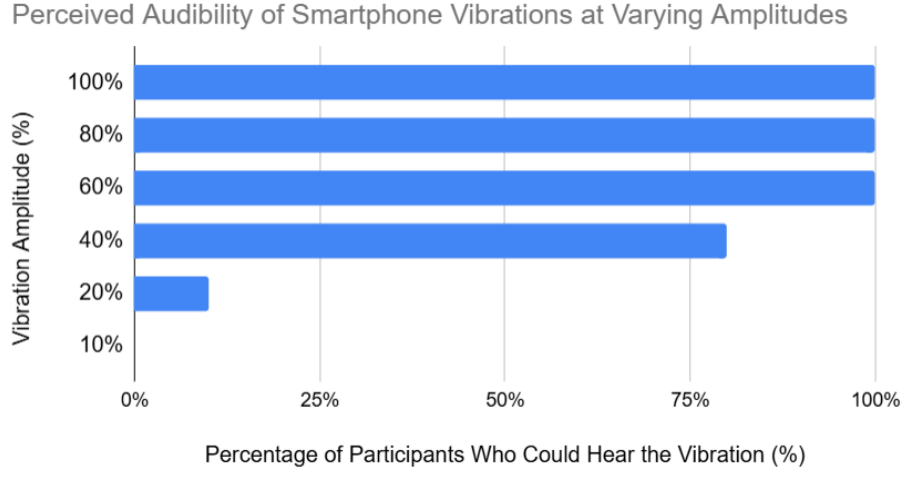
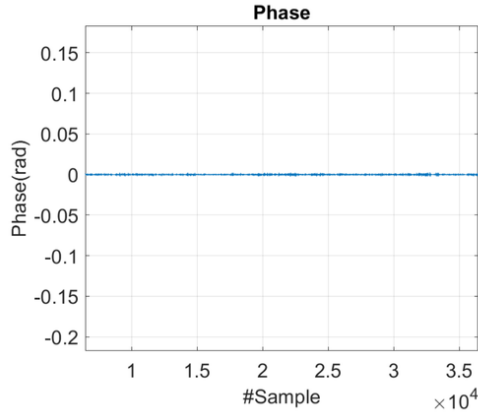
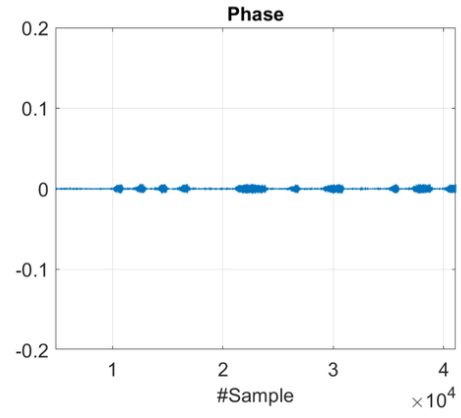


Figure 9: Audibility of different vibration amplitudes



(a) 10% Vibration Amplitude



(b) 20% Vibration Amplitude

Figure 10: Phase variations observed at low vibration amplitudes.

Amplitude Level	BER (%)
20%	0.0
10%	52.5

Table 2: Bit Error Rate at different amplitude levels

1. Minimizing user noticeability
2. Ensuring accurate retrieval using mmWave radar

As shown in Figure 9, an experiment was conducted with 10 participants to evaluate the perceptibility of different vibration amplitudes. The vibration amplitude can be configured with values ranging from 0 to 255. For the experiment, amplitude levels were tested as percentages of this maximum value. The results indicate that vibrations at 10% and 20% of the maximum amplitude were significantly less perceptible compared to higher intensity levels.

To assess data retrieval accuracy at these lower amplitudes, additional tests were performed. A smartphone was placed 1.6 meters from the radar and configured to vibrate in OOK pattern, where a 100 ms vibration represented a binary "1" and no vibration represented a "0". The resulting vibration patterns were captured and analyzed.

The phase variations observed at 10% amplitude, as shown in Figure 10a, were significantly less detectable compared to those at 20% amplitude in Figure 10b, making it more challenging to extract the encoded data. BER% calculations, presented in Table 2, indicate that the BER at 10% amplitude was substantially higher, making data retrieval unreliable. Therefore, 20% amplitude was chosen as the optimal setting for further experiments.

Since the maximum vibration amplitude of the smartphone is 255, a 20% amplitude corresponds to approximately 51. For consistency, an amplitude value of 50 was selected for all subsequent experiments.

4.3 Determining Vibration Period

As explained in Section 3.2.1, the vibration period must be greater than the rise and fall time of the smartphone’s vibration motor, which typically ranges from 4ms to 120ms (Cui et al., 2023). In the smartphone used for this study, it was observed that the mmWave radar detects a rise time of approximately 30 ms for the vibration to reach its maximum amplitude. Therefore, the vibration period must exceed this value to ensure accurate signal capture.

In addition to hardware constraints, the vibration period must be long enough for the mmWave radar to differentiate between the ON and OFF states of the signal. To evaluate this, experiments were conducted with varying vibration periods, and the resulting BER are shown in Figure 11. The results indicate that BER is 0% at both 100ms and 80ms periods. However, BER increases significantly when the period is reduced to 60ms and 40ms.

Although shorter periods allow for higher data rates, they also increase the likelihood of bit errors. Therefore, an 80 ms vibration period was selected for the remaining experiments, as it provides a balance between transmission speed and data reliability.

This 80ms vibration period directly impacts the data rate of the system. The

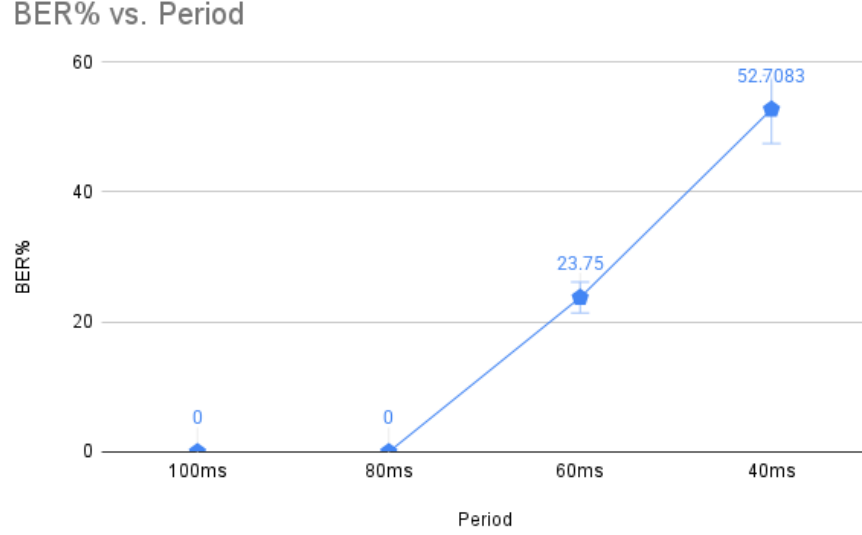


Figure 11: BER% vs Vibration period

theoretical data rate can be calculated using the following formula:

$$\text{Data Rate} = \frac{1}{\text{Bit Duration}}$$

Since each bit is transmitted over 80ms, the data rate would be:

$$\text{Data Rate} = \frac{1}{80 \text{ ms}} = \frac{1}{0.08 \text{ s}} = 12.5 \text{ bits per second (bps)}$$

Therefore, with an 80ms vibration period, the system is capable of transmitting data at a theoretical rate of 12.5 bps. While this rate ensures reliable transmission with minimal bit errors, it is important to note that reducing the vibration period to increase the data rate would lead to a higher likelihood of bit errors, as demonstrated by the experimental results. Thus, the chosen 80ms period provides an optimal trade-off between data transmission speed and accuracy, ensuring the system can covertly transmit data without significant loss of integrity.

4.4 Encoding data

Message Construction and Encoding

The smartphone application extracts sensitive data such as location, credentials, tokens, or clipboard contents and encodes this confidential information into vibration signals.

The encoding begins by converting the sensitive data into a string format, which is then translated into its binary representation using 8-bit ASCII encoding per character. This binary message becomes the basis for further processing.

Before any transmission begins, the app acquires a partial wakelock, ensuring the CPU remains awake even if the screen is off. This allows continuous signal transmission without interruption. To minimize detection and interference, the app also monitors the smartphone's accelerometer sensor. Transmission is triggered only if the phone remains completely stationary for at least 10 minutes, ensuring a stable transmission environment and minimizing the risk of alerting the user.

Frame Structure

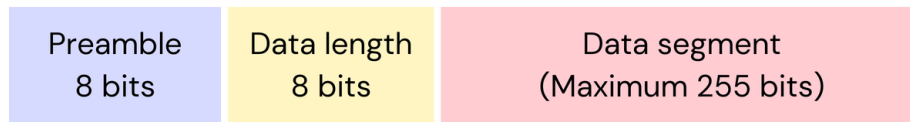


Figure 12: Frame structure with encoded data

The overall frame structure used in the transmission is illustrated in Figure 12. It consists of three main components: a preamble, a length field, and a payload. This structure ensures reliable detection, synchronization, and decoding of the transmitted data. The preamble is a fixed 8-bit sequence (10101010) used for frame detection and synchronization. The following 8-bit length field specifies the size of the payload in bytes. Finally, the payload contains the actual encoded data, which is represented using Hamming (7,4) error-correcting code.

Frame Identification: Preamble Sequence

The preamble sequence 10101010 is placed at the beginning of each transmission frame. It is used for two main purposes:

- Identify the beginning of a new transmission,
- Distinguish between consecutive frames in multi-frame messages.

Its alternating 1s and 0s ensure strong detectability in vibration signals, even under background noise.

Data Length

To enable correct extraction of the payload at the receiver, the transmitter prepends

an 8-bit length field that specifies the number of payload bytes (after Hamming encoding). This allows the receiver to know exactly how many bits to listen for during decoding.

However, since the length field is 8 bits, the maximum value it can represent is 255. If the encoded payload exceeds this limit, the message is automatically split across multiple transmission frames, each with its own preamble and length field. The receiver collects these frames and reassembles them to reconstruct the original sensitive message.

Error Correction: Hamming (7,4) Code

To enhance robustness against bit errors during transmission, the system applies Hamming (7,4) encoding. This transforms every 4-bit block of the binary payload into a 7-bit codeword, with added parity bits for single-bit error detection and correction.

The formula used for Hamming (7,4) encoding is:

- Data bits: d_1, d_2, d_3, d_4
- Parity bits:

$$p_1 = d_1 \oplus d_2 \oplus d_4$$

$$p_2 = d_1 \oplus d_3 \oplus d_4$$

$$p_3 = d_2 \oplus d_3 \oplus d_4$$

This produces a final codeword of the form:

$$[p_1, p_2, d_1, p_3, d_2, d_3, d_4]$$

This process significantly reduces the BER by enabling the receiver to detect and correct transmission errors without needing retransmission.

The result of the error-corrected binary is considered the payload, the actual sensitive content that the attacker aims to exfiltrate.

Covert Transmission via Vibration

The complete binary sequence (preamble + length + payload) is transformed into vibration patterns using a simple, consistent mapping:

- Bit 1: Short vibration (e.g., 80 ms)
- Bit 0: Silent pause (e.g., 80 ms)

These vibration pulses are sent at low amplitude (around 20% maximum amplitude) to avoid user perception while remaining detectable by external mmWave radar receivers.

Data modulation

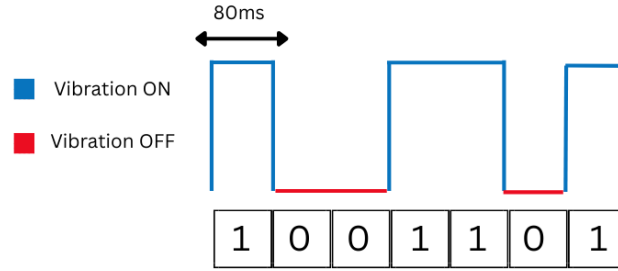


Figure 13: Vibration frame with On Off keying modulation

Data modulation in this context is achieved using OOK, a simple yet effective method for encoding binary data into vibration patterns. In OOK, a "1" bit is represented by a vibration, and a "0" bit is represented by no vibration. Specifically, each "on" state corresponds to a vibration with an amplitude of 50, lasting for 80 milliseconds, while each "off" state is represented by an amplitude of 0 for the same duration, effectively keeping the vibrator off.

The Android API for vibration control allows the generation of vibration patterns by providing different amplitude and duration pairs. For example, the vibration pattern in Figure 13 can be represented as a sequence of time and amplitude pairs like $\langle 80, 50 \rangle$, $\langle 160, 0 \rangle$, $\langle 160, 50 \rangle$, $\langle 80, 0 \rangle$, $\langle 80, 50 \rangle$. Each pair instructs the vibration motor to either vibrate (at the specified amplitude and duration) or remain silent, effectively encoding the message into a series of vibrations.

Encoding and transmission algorithm

Algorithm 4.1 Covert Data Transmission via Vibration

Require: Input text message T

Ensure: Covert transmission using vibration pattern

- 1: Initialize vibrator service and acquire partial wake lock
 - 2: Register accelerometer sensor listener
 - 3: **while** device is moving **do**
 - 4: Wait until no movement is detected
 - 5: Reset movement timer
 - 6: **end while**
 - 7: Start movement timer
 - 8: **if** device remains still for 10 minutes **then**
 - 9: Proceed to encode and transmit data
 - 10: **else**
 - 11: Abort or restart timer
 - 12: **end if**

 - 13: Convert each character in T to 8-bit binary
 - 14: Concatenate all characters into binary string B
 - 15: Apply Hamming(7,4) encoding to $B \rightarrow B_H$
 - 16: Construct final message M :
 - 17: Prepend synchronization preamble (e.g., "10101010")
 - 18: Append length of B_H in 8-bit binary
 - 19: Append B_H

 - 20: Generate vibration pattern P from M :
 - 21: **for** each bit b_i in M **do**
 - 22: **if** $b_i = 1$ **then**
 - 23: Add vibration duration (e.g., 80 ms)
 - 24: **else**
 - 25: Add silence duration (e.g., 80 ms)
 - 26: **end if**
 - 27: **end for**
 - 28: Assign amplitude 50 for vibration, 0 for silence
 - 29: Transmit P using `vibrate(waveform, amplitudes)`
 - 30: Release wake lock after transmission
-

The process of frame preparation and data modulation is outlined in Algorithm 4.1. Initially, the vibrator service is initialized and a partial wake lock is acquired (line 1). The accelerometer sensor listener is registered (line 2) to monitor device movement. If the device is moving, the system waits until no movement is

detected (lines 3–5). After the device remains still for 10 minutes (line 8), data encoding begins (line 9). The input text message is converted into binary (lines 13–14), followed by Hamming(7,4) error correction encoding (line 15). A synchronization preamble and the length of the encoded message are added to construct the final message (lines 16–19). The binary message is then converted into a vibration pattern (lines 20–29), with each bit corresponding to either vibration or silence. Finally, the vibration pattern is transmitted using the vibrator (line 30), and the wake lock is released after transmission (line 31).

4.5 Signal Capture Using mmWave Studio

This research utilizes the mmWave Studio software provided by Texas Instruments which is a Graphical User Interface (GUI) designed for configuring, controlling, and evaluating TI mmWave radar devices. The radar system is initialized and controlled through mmWave Studio by sending low-level Device Firmware Package (DFP) API commands to the radar hardware over an SPI interface.

During operation, the DCA1000 EVM board is used to collect raw ADC data from the radar. This data is then saved in binary format and processed using custom MATLAB scripts these results are displayed using various plots in the GUI (Texas Instruments, 2022).

In this research, the Detection and Angle Estimation graph generated within mmWave Studio (discussed in Section 3.2.2) plays a key role in the initial stage of target identification. This graph displays all detected objects and their estimated distances from the radar. The position of the vibrating mobile phone is identified from this plot and recorded as the expected target range. This estimated location is then used in the MATLAB-based processing code to select the appropriate range bin for extracting the signal associated with the vibrating device.

4.6 Decoding transmitted data

To retrieve the covertly transmitted data from the radar reflections, a structured decoding pipeline was developed. This pipeline consists of three main stages: preprocessing the raw bin file, extracting the target signal, and OOK-based demodulation. The first two stages build upon the mmWave Studio preprocessing code and were further adapted using signal processing methods from the Cui et al. (2023) study, which they used to detect a vibrating target. These components were then extended and refined to align with the specific goals of this research, which focuses on decoding vibration-induced transmissions from a smartphone. The final stage involves interpreting the presence or absence of vibration patterns to recover the binary data transmitted using OOK modulation.

The following sections detail the algorithms and procedures used in each stage of this decoding pipeline, including the specific adaptations made to handle real-time bin file data and the techniques used to ensure accurate signal recovery and decoding.

4.6.1 Preprocessing raw radar data

The preprocessing step in this system is focused on converting raw radar data originally captured in binary format, into a structured and usable format for further analysis. This process deals with binary (.bin) files and the configurations file, which has the configurations used to set up the radar. The program interprets and processes the data using the given configurations.

The preprocessing program in Algorithm 4.2 loads the configuration and setup details (like sampling rate, number of antennas, and chirp settings), validates that the radar and data capture settings are compatible, and then reads the binary file data frame by frame. Each frame is reshaped and converted into a complex number format, which reflects the original signal’s real and imaginary components. Next, a FFT is applied to each frame in the range dimension to produce a structure called the radar cube, which organizes the data in a format suitable for signal processing algorithms. Finally, this processed data is saved in a more convenient .mat format for further use in the receiver or detection pipeline.

In short, Preprocessing transforms raw radar recordings into a cleaned, structured, and frequency-domain format that reflects target position (via range FFT), enabling further analysis or decoding.

4.6.2 Target Detection and Signal Extraction

After preprocessing, the system enters the target extraction phase, where it identifies and isolates the signal containing vibrating signal data from a smartphone. To accurately determine the target’s location, mmWave Studio is used in advance to perform CFAR-based detection as explained in Section 4.5 from this output, the estimated distance of the target is manually noted and used as input to our processing code.

The code, as described in Algorithm 4.3, begins by restructuring the FFT data from every range bin. For each bin, it extracts the complex FFT values across all frames and receiver channels, reshaping them into a matrix where each row represents a time sample (stacked across chirps and frames), and each column represents a receiver antenna. These matrices are stored in the radarCube structure under the variable targetSignal.

After storing the signals for all range bins, the code computes the difference between the input target location and each entry in the radar’s range axis. The range bin with the minimum distance to the estimated target location is selected as the target bin. The signal from this bin is then extracted from targetSignal and

Algorithm 4.2 Radar Data Processing and Export

```
1: for each binary (.bin) file in the Data folder do
2:   Load setup JSON and mmWave configuration file
3:   Validate device and capture configuration
4:   for each .bin file do
5:     Open binary file
6:     Calculate number of frames from file size
7:     for each frame in the file do
8:       Read raw binary data for one frame
9:       Reshape data based on number of lanes (2 or 4)
10:      Swap IQ if required
11:      Convert data to complex format
12:      Arrange into [chirps, channels, samples]
13:      Apply window function (e.g., Blackman)
14:      Perform 1D Range FFT  $\rightarrow$  data_rangeFFT
15:    end for
16:    Export data_rangeFFT to .mat file
      data_rangeFFT dimensions:
       $[N_{chirps}, N_{rangeBins}, N_{Rx}]$ 
      axis 1: Chirps per frame
      axis 2: Range bins (FFT output)
      axis 3: Receiver antennas
17:    Export rest of the metadata to .mat
18:    Close all open files
19:  end for
20: end for
```

saved in the variable `signal`. This extracted signal serves as the input for further processing.

Algorithm 4.3 Target Detection and Signal Extraction

```

1: Input: Preprocessed .mat files containing data_rangeFFT, Target location from
   mmWave Studio CFAR based plot
2: Output: Extracted target signal vibSig
3: for each .mat file in the dataset do
4:   Load file and initialize radarCube
5:   if radarCube is empty then
6:     Break with error
7:   end if
8:   Extract range axis: rangeFFT_x = radarCube.rfParams.rangeFFT_x
9:   Reformat and save FFT data for each range bin:
10:  for each range bin index idx = 1 to numRangeBins do
11:    for each frame frameIdx = 1 to NFrame do
12:      for each Rx channel chIdx = 1 to NRxChan do
13:        Extract FFT data for rangeIdx, store in DataArray
14:      end for
15:    end for
16:    for each Rx channel chIdx = 1 to NRxChan do
17:      Unroll 2D DataArray into 1D vector over time
18:    end for
19:    Save result in radarCube.targetSignal(:, :, idx)
20:  end for
21:  Identify closest range bin to the target location
22:  Extract signal from identified bin
23:  vibSig = radarCube.targetSignal(:, :, idx)
24:  output vibSig
25: end for

```

4.6.3 Signal Processing and Demodulation

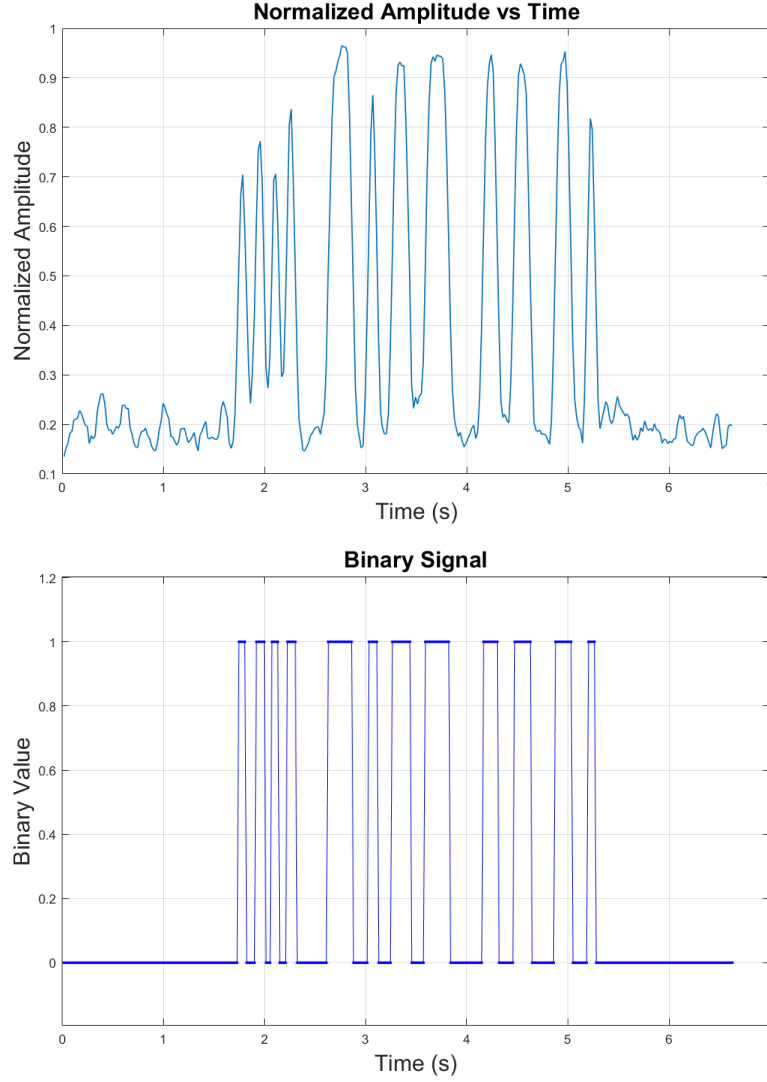


Figure 14: Top: Normalized amplitude graph Bottom: Binary signal generated using thresholding

The radar receives complex signals from multiple antennas. The goal is to isolate phase variations caused specifically by the vibrating smartphone. The first three steps in Algorithm 4.4 depict phase isolation. First, Beamforming is applied to focus the signal reception from a particular direction using a specified angle. In this study, a steering angle of 0 degrees (converted to radians internally) is used to focus

reception along the direct line-of-sight path since the phone is expected to be directly in front of the radar. Once beamforming is applied, the phase of the resulting signal is computed and adjusted to maintain continuity over time. A bandpass filter is then applied to this phase signal, targeting the expected vibration frequency range of 100–300 Hz. This filtering isolates periodic phase changes associated with the vibration signal while attenuating irrelevant components.

Next, the filtered phase signal is processed using a spectrogram to visualize and analyze amplitude variation over time. The amplitude is smoothed and normalized, and a threshold-based detector is applied (line 10 - 17). The threshold is set at 50% of the maximum normalized amplitude based on the assumption that the dominant phase variation within the filtered band arises from the object’s intentional vibration. Therefore, if the amplitude of a frequency component exceeds this threshold, it is likely due to a real vibration rather than noise or background motion. This thresholding process generates a binary signal as shown in Figure 14 representing the ‘on’ and ‘off’ states of the vibrating source, forming the basis for On-Off Keying (OOK) demodulation.

A known preamble pattern (e.g., 10101010) is used for synchronization and is located in the binary stream via cross-correlation. Immediately following the preamble, 8 bits are used to encode the length of the actual data payload. These bits are extracted using the `getLength()` function (line 21-22), which interprets them using OOK decoding by measuring pulse widths. A pulse must last at least 50% of the expected bit duration (80 ms) to be considered a valid binary ‘1’ or ‘0’. After determining the payload length, the `extractBinaryFromSignal()` function (line 23 - 31) scans the binary signal chunk by chunk, matching each with idealized OOK pulses to recover the corresponding bits. The extracted binary bits, shown in Figure 15, clearly reflect the transmitted frame structure. Finally, the payload is passed through a Hamming (7,4) decoder for error correction, and the resulting corrected binary stream is converted to text.

All implementation code used in this work can be found in the GitHub repository link : <https://github.com/Dhatchaya/research>

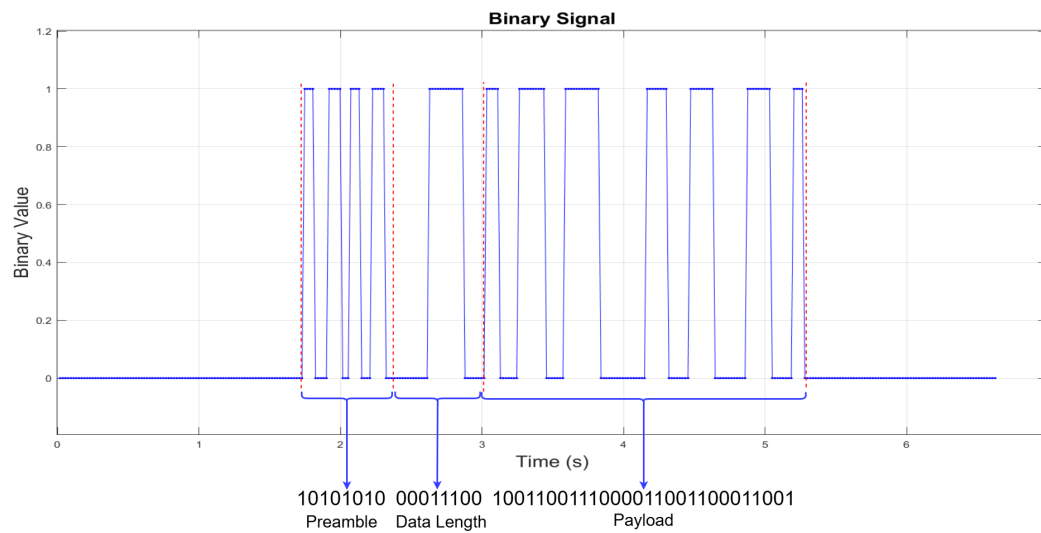


Figure 15: Binary signal containing transmitted frame structure

Algorithm 4.4 On-Off Keying (OOK) Decoding Algorithm

Require: Complex radar signal S , vibration frequency range [100–300] Hz, steering angle $\theta = 0^\circ$, pulse width = 80 ms, amplitude threshold θ , Preamble pattern P

Ensure: Decoded binary message M

- 1: **Step 1: Beamforming and Phase Extraction**
 - 2: Apply beamforming to S using angle $\theta = 0^\circ$ to enhance line-of-sight reception.
 - 3: Extract and unwrap the phase of the beamformed signal to ensure continuity.
 - 4: **Step 2: Bandpass Filtering**
 - 5: Apply a bandpass filter to the unwrapped phase signal within 100–300 Hz.
 - 6: **Step 3: Spectrogram and Amplitude Tracking**
 - 7: Compute a spectrogram of the filtered phase signal.
 - 8: Sum across frequency bins to get amplitude vs. time
 - 9: Smooth and normalize the amplitude to obtain signal A .
 - 10: **Step 4: Binarization of Signal**
 - 11: **for** each amplitude sample A_i in A **do**
 - 12: **if** $A_i > \theta$ **then**
 - 13: $B_i \leftarrow 1$
 - 14: **else**
 - 15: $B_i \leftarrow 0$
 - 16: **end if**
 - 17: **end for**
 - 18: **Step 5: Preamble Detection**
 - 19: Compute cross-correlation $C = B * P$
 - 20: Find peak index t_{start} where C is maximum
 - 21: **Step 6: Extract Data Length**
 - 22: Extract first 8 bits after $t_{start} + |P|$ as length field and Convert to decimal L
 - 23: **Step 7: Extract Binary From Signal:**
 - 24: **for** $i = 1$ to L **do**
 - 25: Extract signal chunk corresponding to one pulse
 - 26: **if** chunk matches ideal '1' pulse **then**
 - 27: Append '1' to data
 - 28: **else if** chunk matches ideal '0' pulse **then**
 - 29: Append '0' to data
 - 30: **end if**
 - 31: **end for**
 - 32: **Step 8: Hamming(7,4) Error Correction**
 - 33: **for** each 7-bit block in binary data **do**
 - 34: Correct errors using Hamming(7,4) decoding and Extract 4-bit message
 - 35: **end for**
 - 36: **Return:** Final decoded message M
-

5 Evaluation and Results

The evaluation of our system was conducted from two primary perspectives: the stealthiness of the covert channel and the effectiveness of using mmWave radar to exploit this channel. To assess the undetectability of the vibration-based covert communication, we performed qualitative tests involving multiple human participants, focusing on whether the transmitted vibrations could be perceived across various surfaces and amplitude levels. Meanwhile, the technical feasibility of detecting and decoding the vibrations using mmWave radar was evaluated through a series of controlled experiments using a hardware setup described in Section 4.1. This dual evaluation approach enables us to demonstrate both the covert nature of the transmission and the practicality of remote data extraction using radar technology.

5.1 Evaluation matrix

To assess the performance and stealthiness of the proposed covert communication system, we rely on a combination of quantitative and qualitative evaluation metrics. These metrics provide a comprehensive understanding of both the technical reliability and the perceptual detectability of the channel.

5.1.1 Bit Error Rate (BER)

The primary quantitative metric used in our evaluation is the BER. It reflects the accuracy of data transmission and decoding using the mmWave radar system. BER is defined as the ratio of incorrectly received bits to the total number of transmitted bits and is calculated using the following equation:

$$\text{BER (\%)} = \left(\frac{N_{\text{errors}}}{N_{\text{total bits}}} \right) \times 100 \quad (12)$$

where N_{errors} represents the number of bit errors detected at the receiver, and $N_{\text{total bits}}$ is the total number of bits transmitted. A lower BER indicates higher system reliability. This metric was used to evaluate decoding success under various experimental conditions such as surface type, distance, and device orientation.

5.1.2 Data Rate

The data rate is a measure of how quickly information can be transmitted and is expressed in **bits per second (bps)**. It is calculated using the formula:

$$\text{Data Rate (bps)} = \frac{1}{T_{\text{bit}} + T_{\text{delimiter}}}$$

Here, T_{bit} represents the time taken to transmit a single bit, and $T_{\text{delimiter}}$ is the duration of any gap or separator added between bits or symbols. The delimiter is included only if the encoding scheme in use defines such a separation. Both values are measured in seconds.

This metric is important for evaluating how efficient a given encoding strategy is in transmitting data within the covert channel.

5.1.3 Participant-Based Stealth Evaluation

To evaluate the stealthiness of the covert channel, we conducted a qualitative study involving participant feedback. During the experiments, 10 participants were exposed to vibration signals of varying amplitudes and different surface materials. They were asked to report whether they could hear the vibrations.

The responses were aggregated and converted into percentages, indicating the perceptibility of the vibration signals across different conditions. These values serve as a qualitative metric to assess the covert nature of the transmission. A lower detection rate among participants suggests a higher level of stealth, supporting the viability of the channel for covert communication.

5.2 Vibration Audibility Across Different Surface Materials

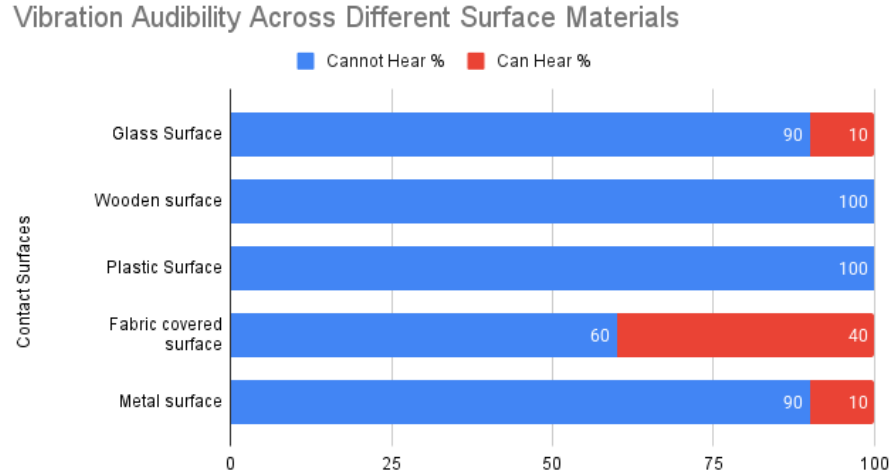


Figure 16: Vibration audibility perception across different surface materials

To evaluate the perceptibility of vibration-based transmission, we conducted a user study involving 10 participants from various age groups. The distribution of

Age Group	Number of Participants
10–20	3
20–30	4
40–50	1
50–60	1
60–70	1

Table 3: Participant distribution across different age groups

participants across different age groups is shown in Table 3. This diverse age representation was aimed to capture potential variation in vibration perception due to age-related sensory sensitivity.

The experiment involved playing a fixed vibration pattern through our Android application, with the vibration amplitude set to 50 (on a scale from 0 to 255). The smartphone was placed on different surface materials, namely, glass, wood, plastic, fabric covered surfaces (e.g., Sofa, fabric-covered iron tables), and metal. Participants were asked to indicate whether they could hear the vibrations emitted through each surface.

The results, summarized in Figure 16, show a clear variation in perceptibility across different materials. Plastic and wooden surfaces had the highest stealth performance, with 0% of participants reporting any audible feedback. Glass and metal surfaces showed slightly more leakage, with only 10% of participants detecting the vibration. In contrast, the fabric surface had a significantly higher audibility rate, with 40% of participants reporting that they could hear the vibration.

This unexpectedly high audibility on fabric surfaces is likely due to the nature of the fabric used and the structural support underneath. In our experiments, the fabric was not loosely draped, but rather firmly stretched over underlying materials, such as sofa cushions or tabletops, which are commonly encountered in real-world settings like office furniture or home couches. These setups were intentionally chosen to reflect practical usage scenarios, where smartphones are often placed on upholstered or fabric-covered surfaces.

Such firm fabric coverings can transmit and even amplify vibrations, especially when paired with resonant support structures. Unlike soft or fluffy fabric that typically dampens sound, these real-world surfaces can act similarly to a membrane-resonator system, making the vibrations more perceptible.

These findings indicate that both the surface material and its structural context play a critical role in the stealthiness of vibration-based transmissions. Materials like plastic and wood are more effective in masking vibration noise due to their non-resonant, energy-absorbing properties, making them ideal candidates for covert communication scenarios. In contrast, seemingly soft materials like fabric can actu-

ally reduce stealth if coupled with resonant structures, highlighting the importance of evaluating the surface as a whole.

5.3 Evaluation of Different Encoding Schemes

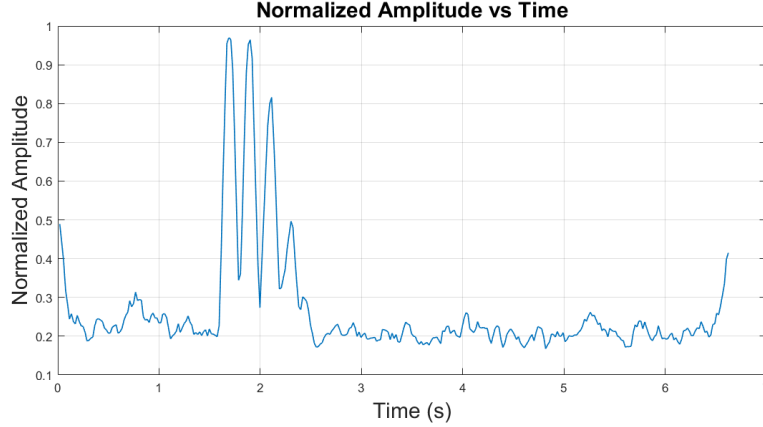


Figure 17: Amplitude vs. time representation of the vibration levels used in PWAM

In this work, we evaluate two different vibration-based encoding schemes to determine their effectiveness in transmitting data using a smartphone’s vibration motor. The first method is OOK, where each bit is represented simply by either the presence or absence of vibration for a fixed duration. This method is straightforward and is described in detail in Section 4.4. The second method is a more advanced approach we refer to as Pulse Width and Amplitude Modulation (PWAM). In this scheme, we encode information using different vibration amplitudes and fixed durations, allowing us to represent multiple bits per symbol. In this scheme, information is encoded using vibration amplitude levels, where each amplitude represents 2 bits. Specifically, we use four amplitude levels: 80, 60, 40, and 20, corresponding approximately to 31%, 23%, 15%, and 8% of the maximum amplitude, each mapped to a 2-bit binary combination. These levels were chosen based on the perceptibility and stability analysis discussed in Section 4.2, where we identified them as the most distinguishable and less noticeable amplitudes. Figure 17 shows the amplitude vs time graph of these four amplitude levels. Each amplitude level is maintained for a fixed duration of 100ms, which, as explained in Section 4.3, helps ensure a lower BER by giving the motor sufficient time to fully stabilize at the desired amplitude.

To further support this stabilization and to reduce inter-symbol interference, especially for lower amplitude levels like 8%, we add a gap between each symbol. Instead of leaving a long silent gap or switching suddenly, we first include 50ms of no vibration, followed by 50ms of very weak vibration at just 4% of the maximum

amplitude. This gentle vibration helps keep the motor slightly active, so it doesn't have to start from complete rest. As we observed in Section 3.2.1, the vibration motor needs at least 30ms to reach its full strength. Because of this, even low amplitudes can quickly reach their full strength when the actual symbol begins. This approach improves the accuracy of decoding by making the vibrations more consistent and reliable.

5.3.1 Data frame structure

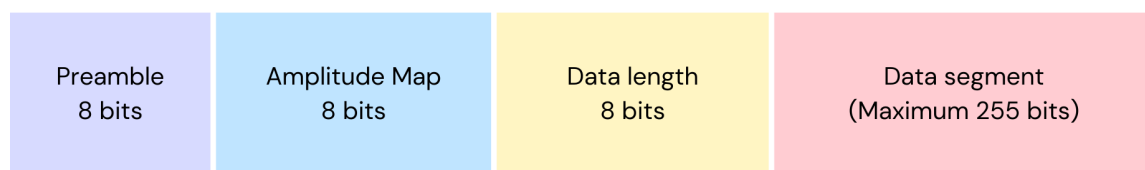


Figure 18: Frame structure with Pulse Width Amplitude Modulated data

The complete structure of a transmission frame using the PWAM method is carefully designed to ensure reliable communication and accurate decoding. As illustrated in Figure 18, the frame begins with a preamble, which is a fixed alternating high-low pattern 10101010 using four 80 amplitude level symbols with delimiter to signal the start of a transmission. It helps the receiver synchronize and prepare for decoding the actual data. Following the preamble, we include a series of calibration pulses. These are fixed 100 ms pulses at amplitude levels of 80, 60, 40, and 20. These are used by the receiver to calibrate thresholds for detecting data symbols.

Next, we transmit 4 amplitude-modulated symbols (2 bits each) to represent data length. This field tells the receiver how many bits or symbols to expect in the upcoming payload, encoded using the same amplitude modulation approach as the actual data. Finally, the payload section contains the actual message, encoded using 2-bit per symbol modulation. Each symbol is 100 ms long, followed by a 50ms no vibration and 50ms of low intensity vibration between each symbol.

Together, this structured approach ensures that the receiver can reliably detect the beginning of a message, calibrate the amplitude thresholds, determine the expected message length, and accurately decode the data all based on vibration alone.

5.3.2 Encoding Data into Vibration Signal

The PWAM Encoding and transmission Algorithm 5.1 begins by converting the input text into binary (line 2) and applying Hamming(7,4) error correction encoding (line 3). A vibration frame is then constructed using a fixed preamble (line 4), a predefined amplitude header with 4 levels of amplitude for calibration (line 5), and

a length field indicating the size of the encoded data (line 6). These components are concatenated with the encoded payload to form the complete frame (line 7), which is then transformed into a vibration pattern (line 8) and transmitted via the vibration motor (line 9). The helper function `ConvertTextToBinary` (lines 11–17) encodes each character of the text into its 8-bit binary representation. The `ApplyHamming74` function (lines 18–26) encodes each 4-bit chunk of the binary data using Hamming(7,4) coding to ensure error detection and correction. The `GeneratePattern` function (lines 27–36) maps each 2-bit symbol from the binary frame to one of four predefined amplitudes 80,60,40 or 20 and introduces structured pauses between pulses to reduce inter-symbol interference. Finally, the `Vibrate` function (lines 37–39) triggers the vibration motor using the generated pattern.

5.3.3 Decoding Amplitude Modulated Vibration signal

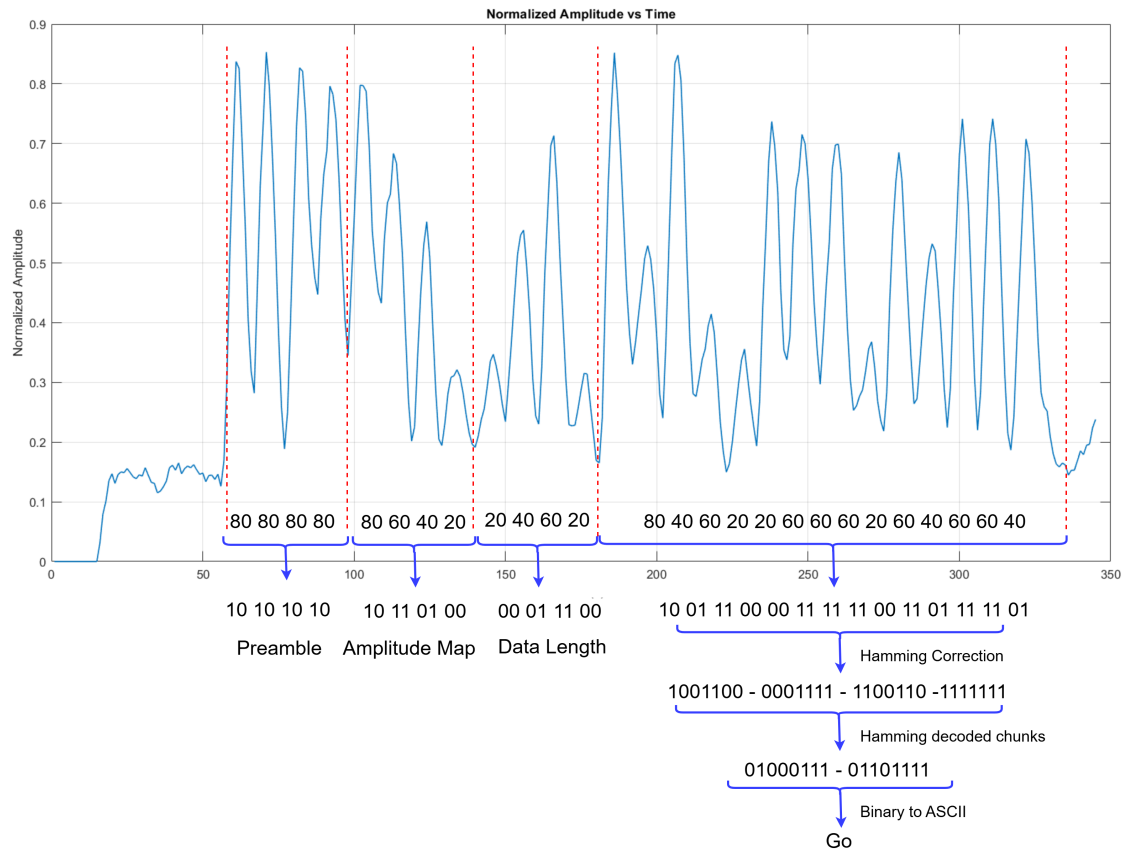


Figure 19: Demodulation

Algorithm 5.1 PWAM Encoding and transmission

```
1: function STARTVIBRATION(text)
2:   binaryData  $\leftarrow$  ConvertTextToBinary(text)
3:   encodedData  $\leftarrow$  ApplyHamming74(binaryData)
4:   preamble  $\leftarrow$  "10101010"
5:   amplitudeHeader  $\leftarrow$  amplitudes (80,60,40,20)
6:   lengthField  $\leftarrow$  8-bit binary of length(encodedData)
7:   frame  $\leftarrow$  preamble || amplitudeHeader || lengthField || encodedData
8:   vibrationPattern  $\leftarrow$  GeneratePattern(frame)
9:   Vibrate(vibrationPattern)
10: end function
11: function CONVERTTEXTTOBINARY(text)
12:   result  $\leftarrow$  empty string
13:   for each character  $c$  in text do
14:     append 8-bit binary of  $c$  to result
15:   end for
16:   return result
17: end function
18: function APPLYHAMMING74(binaryData)
19:   result  $\leftarrow$  empty string
20:   for each 4-bit chunk in binaryData do
21:     pad chunk to 4 bits if needed
22:     encoded  $\leftarrow$  Hamming encoded string of each chunk
23:     append encoded to result
24:   end for
25:   return result
26: end function
27: function GENERATEPATTERN(binaryFrame)
28:   Initialize empty patternList and amplitudeList
29:   for each 2-bit symbol in binaryFrame do
30:     Map symbol to amplitude:
31:     "00"  $\rightarrow$  20, "01"  $\rightarrow$  40, "10"  $\rightarrow$  80, "11"  $\rightarrow$  60
32:     Add 100ms pulse with mapped amplitude
33:     Add 50ms silence and 50ms low amplitude (10) as separator
34:   end for
35:   return VibrationPattern(patternList, amplitudeList)
36: end function
37: function VIBRATE(pattern)
38:   Trigger vibration motor with timing and amplitude arrays
39: end function
```

Decoding Algorithm 5.2 starts by locating the preamble in the amplitude signal using `findPreamble()` (Line 2), which helps identify where the actual message begins. It then extracts four pulses immediately after the preamble for calibration (Line 3), computes their average amplitudes, and uses them to set threshold values between amplitude levels (Line 4). These thresholds are then used to decode the next four pulses, which contain the binary-encoded data length (Line 5). The remaining signal is processed to extract average amplitudes for each pulse (Line 6), which are classified into one of four levels 80, 60, 40, or 20 using the previously set thresholds (Line 7). Each amplitude is mapped to a 2-bit binary value (Line 8), forming the raw binary stream. This stream is then decoded using Hamming(7,4) error correction (Line 9), and the corrected bits are compared against a reference string to compute the Bit Error Rate (BER) (Line 10). Finally, the decoded binary is converted into ASCII characters to reconstruct the original message (Line 11). This Demodulation procedure is further illustrated in Figure 19

Algorithm 5.2 Decoding Amplitude Modulated Vibration Signal

```

1: function DECODEVIBRATIONSIGNAL( $A$ )
2:   Find preamble using FINDPREAMBLE().
3:   Extract 4 calibration pulses after the preamble.
4:   Compute average amplitudes of the pulses and set the thresholds.
5:   Estimate data length using 4 subsequent pulses and decode their binary value.
6:   Read the remaining signal and extract pulse average amplitudes.
7:   Classify amplitudes into four levels : {80, 60, 40, 20} based on the thresholds.
8:   Map amplitude pattern to binary using:
      • 80  $\rightarrow$  10
      • 60  $\rightarrow$  11
      • 40  $\rightarrow$  01
      • 20  $\rightarrow$  00
9:   Decode binary stream using Hamming(7,4) decoding.
10:  Compare decoded bits with a reference string and compute BER.
11:  Convert final binary bits to ASCII text.
12: end function

```

5.3.4 Comparison of OOK and PWAM

Both the encoding schemes were tested at a fixed distance of 1 meter between the smartphone and the mmWave radar. As shown in Table 4, PWAM exhibits a lower data rate and a higher BER compared to OOK. While the data rate of PWAM

can potentially be improved by reducing the symbol duration, this may negatively impact the radar’s ability to distinguish between different amplitude levels during signal processing. As a potential avenue for future work, machine learning-based models could be explored to more effectively recognize amplitude patterns and enhance decoding performance, thereby improving both data rate and accuracy under the PWAM scheme.

Table 4: Comparison of OOK and PWAM encoding Schemes

Scheme	Data Rate (bps)	BER (%)	Remarks
OOK	12.5	0.00	Reliable, higher throughput
PWAM	10	13.75	Lower throughput, higher error

5.4 Evaluation Based on Distance

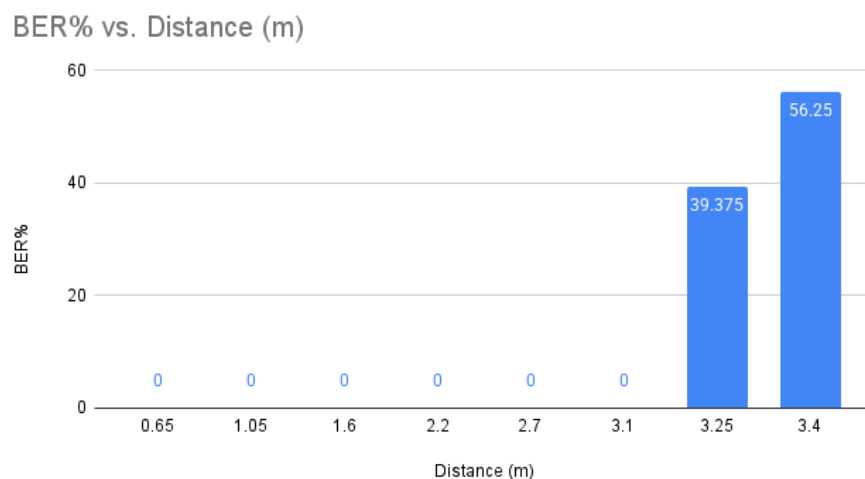
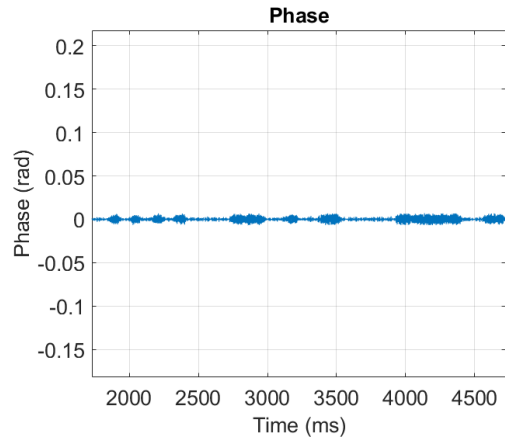


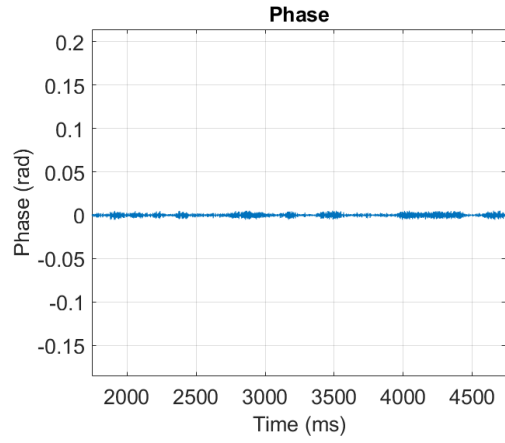
Figure 20: Vibration decodability at different distances

To evaluate the impact of distance on the system’s performance, a set of controlled experiments was conducted by placing the smartphone at varying distances from the mmWave radar. The device was held steadily using a phone holder, with its front consistently facing the radar to ensure a direct line of sight. For each distance, 10 vibration samples were transmitted using a fixed pattern, and the BER was calculated based on the decoding accuracy.

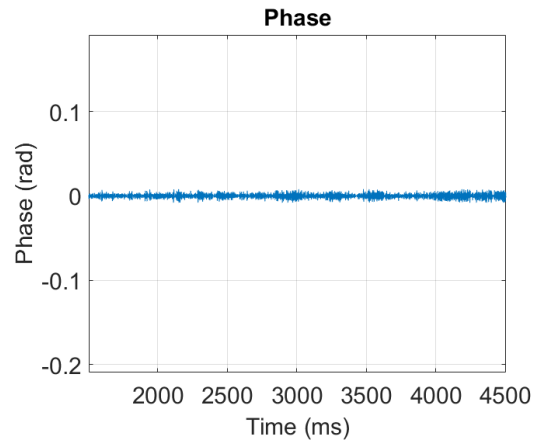
The results are presented in Figure 20, which illustrates the BER percentage relative to distance. Up to 3.1 meters, the system maintained flawless performance,



(a) Phase at 3.1m



(b) Phase at 3.25m



(c) Phase at 3.4m

Figure 21: Phase variation of received signal at increasing distances.

achieving a 0% BER across all samples. However, beyond this range, there was a sharp increase in errors. At 3.25 meters, the BER rose to 39.375%, and further increased to 56.25% at 3.4 meters.

This drop in performance is likely due to the weakening of the reflected mmWave signals as the distance increases. As can be seen in Figure 21 when distance increases, the signal strength weakens, making the phase variations become less clear and distinguishable. As a result, the system’s ability to detect and decode vibrations deteriorates significantly beyond the 3.1 meter mark.

However, for the purposes of a covert communication channel, a reliable range of up to 3 meters is often sufficient, especially in real-world scenarios such as desks, meeting rooms, or other close-range indoor environments. This highlights the practical feasibility of the proposed method in stealthy data exfiltration attacks within confined spaces.

5.5 Impact of Barriers on Signal Propagation

To evaluate how different barriers affect the performance of the covert communication channel, we conducted an experiment by placing various barrier materials between the mmWave radar and the smartphone. The smartphone was positioned at a fixed distance of 0.55 meters from the radar, and the barrier was placed 0.39 meters away from the radar i.e., between the two devices.

The goal was to assess how the presence and type of barrier influence the bit error rate (BER) in vibration signal decoding. All tested barrier materials were approximately 5 mm thick or less, to simulate real-world obstructions like folders, thin boards, or windows.

Table 5: BER for Different Cardboard Thicknesses

Thickness (mm)	BER (%)
2	3.75
5	41.875

The results, summarized in Figure 22, show a clear degradation in system performance as barrier materials are introduced.

These findings can be attributed to the mmWave signal attenuation and reflection properties of different materials. Plastic and glass, due to their higher density and smooth surface, tend to reflect and absorb mmWave signals more than porous and lightweight materials like cardboard. As a result, fewer signal components reach the smartphone, reducing the system’s ability to track phase variations reliably.

As shown in Figure 23, the phase variation plots under different barrier conditions reveal that although some vibration patterns are faintly visible, they are not

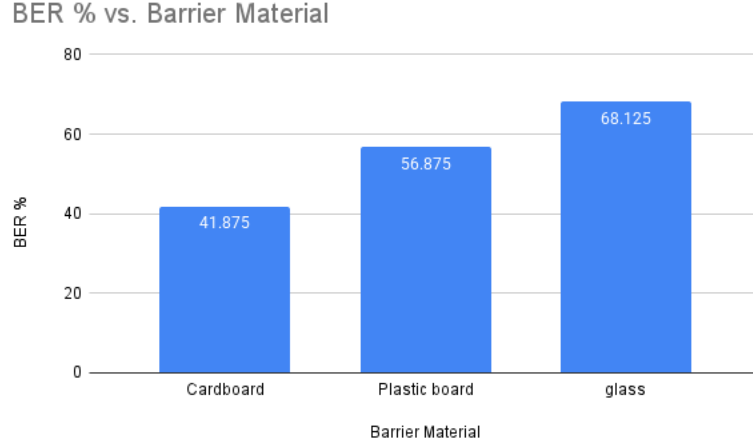


Figure 22: BER for Different Barrier Materials

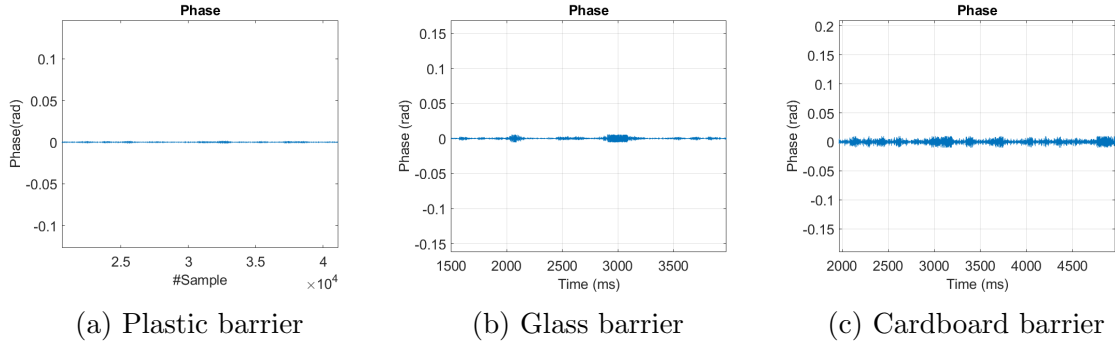


Figure 23: Phase variations observed when different barrier materials were placed between the radar and the smartphone.

consistently clear or distinguishable enough for accurate decoding. This suggests that the barriers interfere with the reflected mmWave signals, causing the signal to fluctuate or weaken. These disruptions make it increasingly difficult for the system to detect and interpret the transmitted vibrations.

To further validate the influence of barrier thickness, an additional experiment was conducted with two cardboard sheets of different thicknesses: 2 mm and 5 mm. The thinner cardboard led to a much lower BER of 3.75%, whereas the thicker one caused a significantly higher BER of 41.875%. These results support the hypothesis that thicker materials absorb and scatter more mmWave energy, reducing signal strength and causing phase instability.

These results indicate that the system is only able to tolerate barriers with minimal thickness, and even then, the material composition plays a critical role in deter-

mining performance. While thinner barriers like 2mm cardboard result in negligible degradation, thicker or more signal-reflective materials such as plastic or glass significantly weaken the mmWave reflections, leading to high BER. Therefore, for this covert communication system to remain effective in real-world scenarios, barriers must be both thin and composed of materials that minimally attenuate or scatter mmWave signals.

5.6 Evaluation with Different Smartphone Models

This evaluation was conducted to assess the system’s performance across multiple generations of smartphones, specifically different models of Google Pixel devices. These devices were chosen because they support amplitude control for vibration at the code level, which is essential for maintaining consistency in the experiment.

The test was performed using Pixel 3, Pixel 5, Pixel 7 Pro, Pixel 8 Pro, and Pixel 9. Each device was placed individually at a fixed distance of 0.64 m from the millimeter wave radar using a phone holder to ensure a consistent and stable position directly in front of the radar. For each phone, 10 samples were tested, and the average BER was calculated to assess the performance of the system across different models.

The objective of this evaluation was to generalize the findings across multiple Pixel models ranging from early to the most recent releases. Despite differences in Android versions, dimensions, internal component layout, build materials, and software-level haptic tuning, the system maintained a 0% BER across all tested models, confirming that the vibration signal was reliably detectable and decodable regardless of device generation. The results are summarized in Table 6.

Table 6: BER Performance across Different Pixel Smartphone Models

Smartphone Model	BER (%)
Pixel 3	0.00
Pixel 5	0.00
Pixel 7 Pro	0.00
Pixel 8 Pro	0.00
Pixel 9	0.00

6 Conclusion and Future Work

This research explored the feasibility of using a mmWave FMCW radar to detect smartphone vibrations for covert data transmission. Through a series of experiments, the study investigated the effectiveness of the radar in sensing vibration patterns generated by the smartphone, and how these patterns could be used to encode

and transmit information without alerting the user. The system was tested across different surfaces, distances, barriers, orientations, and phone models to assess both its stealth and robustness.

6.1 Conclusion for Research Question 1

To what extent does a mmWave FMCW radar detect smartphone vibrations?

The results demonstrate that mmWave radar can reliably detect smartphone vibrations at distances up to approximately 3.1 meters in an unobstructed environment. The radar consistently achieved a 0% BER up to this range, with performance degrading sharply beyond 3.25 meters. Additionally, the radar successfully detected vibrations even when common household barriers, such as cardboard, plastic, and glass, were placed between the device and the radar although the material and thickness of these barriers significantly influenced decoding performance. Lightweight materials like thin cardboard were more transparent to vibrations, while denser materials introduced higher error rates.

These results confirm the hypothesis that the radar’s ability to detect vibrations is influenced by several factors, including signal strength, vibration period, distance, and barriers. However, the use of optimized signal processing techniques and well-calibrated transmission parameters enables the radar to accurately decode data from subtle vibration signals under a wide range of conditions.

6.2 Conclusion for Research Question 2

What methodology can be used to effectively transfer data from a smartphone covertly into a mmWave FMCW radar using vibration patterns?

To effectively and covertly transfer data from a smartphone into a mmWave FMCW radar using vibration patterns, we designed a methodology centered on minimizing human perceptibility while ensuring reliable radar detection. The system was developed and tested in real-world environments to validate its stealthiness and performance.

First, a detailed study was conducted to identify the maximum vibration amplitude that remains imperceptible to humans and tested across various surface types. This helped define a safe amplitude range that could be used for data encoding without drawing attention. Different vibration periods were then experimented with, and a 80ms symbol duration was selected as it provided enough time for the radar to accurately detect the signal while keeping the transmission subtle.

To further support covert transmission, we evaluated multiple encoding schemes and ultimately chose OOK due to its low BER and high data rate, which allows ef-

ficient transmission without requiring large or noticeable amplitude changes. The scheme also avoids the need for complex amplitude differentiation, reducing the chances of detection.

Our methodology was validated through experiments under different device orientations and with thin barriers placed between the smartphone and radar, showing that covert communication remains feasible even in non-ideal setups.

In summary, by carefully selecting vibration amplitudes, symbol timing, and encoding strategies, we demonstrated a reliable and stealthy data transmission channel from a smartphone to a mmWave radar one that can operate covertly in everyday environments without alerting nearby individuals.

6.3 Conclusion for Research Problem

The research successfully demonstrates that mmWave FMCW radar can be used as a covert receiver for vibration-based data exfiltration from smartphones. While prior studies had focused on using radar for sensing predefined vibration patterns, this study uniquely addressed the challenge of decoding dynamic data encoded through reduced vibration amplitude and duration, without alerting the user.

The evaluation shows that the proposed system is stealthy, hardware-agnostic (across different Pixel devices), resilient across various orientations and ensures covert nature across material surfaces. These findings open up new possibilities in the domain of cyber-physical security, highlighting a previously unexplored threat vector that leverages existing radar technology for covert data transmission.

6.4 Limitations and Future Directions

While the proposed system demonstrates the feasibility of covert data transmission from a smartphone to a mmWave FMCW radar using vibration patterns, several limitations exist that present avenues for future research.

- The current evaluation only considers a line-of-sight scenario where the smartphone is placed directly facing the radar, corresponding to a 0-degree angle. However, in real-world use cases, the phone may be placed at various angles relative to the radar. Future studies can explore the impact of different angular placements on the accuracy and reliability of the received signal.
- To preserve the covert nature of the communication, PWAM was tested using low-intensity vibration levels. While this helps in making the transmission less noticeable to human perception, it introduces challenges in distinguishing between closely spaced amplitude levels using conventional signal processing

methods. Machine learning techniques could be employed to identify subtle patterns and improve decoding accuracy.

- The energy consumption of generating vibration patterns was not analyzed in this study. Investigating the energy cost associated with different vibration patterns and optimizing for lower power usage could make the system more practical for long-term deployment.
- This study was conducted using only Android smartphones, specifically different models of Google Pixel devices. While these results provide a good baseline, testing the system on a wider range of devices, including other Android brands and possibly iOS devices, would help generalize the findings. Devices with modifiable vibration controls at the firmware level could offer further optimization opportunities.
- The current setup assumes a single target scenario. Future enhancements can extend the system to handle multiple targets simultaneously and implement mechanisms for separating and decoding data from multiple vibrating sources in the radar's field of view.

References

- Ahmed Al-Haiqi, Mahamod Ismail, and Rosdiadee Nordin. A new sensors-based covert channel on android. *The Scientific World Journal*, 2014, 09 2014. doi: 10.1155/2014/969628.
- Bushra Aloraini, Daryl Johnson, Bill Stackpole, and Sumita Mishra. A new covert channel over cellular voice channel in smartphones. 04 2015.
- Suryoday Basak and Mahanth Gowda. mmspy: Spying phone calls using mmwave radars. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1211–1228, 2022. doi: 10.1109/SP46214.2022.9833568.
- Rosangela Casolare, Fabio Martinelli, Francesco Mercaldo, and Antonella Santone. Colluding covert channel for malicious information exfiltration in android environment. In *International Conference on Information Systems Security and Privacy*, 2021. URL <https://api.semanticscholar.org/CorpusID:232106134>.
- Swarup Chandra, Zhiqiang Lin, Ashish Kundu, and Latifur Khan. Towards a systematic study of the covert channel attacks in smartphones. volume 152, pages 427–435, 11 2015. ISBN 978-3-319-23828-9. doi: 10.1007/978-3-319-23829-6_29.
- Kaiyan Cui, Qiang Yang, Yuanqing Zheng, and Jinsong Han. mmripple: Communicating with mmwave radars through smartphone vibration. In *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks*, IPSN '23, page 149–162, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701184. doi: 10.1145/3583120.3586956. URL <https://doi.org/10.1145/3583120.3586956>.
- Kaiyan Cui, Leming Shen, Yuanqing Zheng, Fu Xiao, and Jinsong Han. Talk2radar: Talking to mmwave radars via smartphone speaker. In *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, pages 2358–2367, 2024. doi: 10.1109/INFOCOM52122.2024.10621296.
- Luke Deshotels. Inaudible sound as a covert channel in mobile devices. In *Proceedings of the 8th USENIX Conference on Offensive Technologies*, WOOT'14, page 16, USA, 2014. USENIX Association.
- Wade Gasior and Li Yang. Network covert channels on the android platform. *Proc. 7th Annu. Workshop CSIIRW, Oak Ridge, TN, USA*, 10 2011. doi: 10.1145/2179298.2179367.
- Texas Instruments. *User's Guide: DCA1000EVM Data Capture Card*, 2019. Accessed: 2025-04-22.
-

- Chengkun Jiang, Junchen Guo, Yuan He, Meng Jin, Shuai Li, and Yunhao Liu. mmvib: micrometer-level vibration measurement with mmwave radar. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370851. doi: 10.1145/3372224.3419202. URL <https://doi.org/10.1145/3372224.3419202>.
- Ed Novak, Yutao Tang, Zijiang Hao, Qun Li, and Yifan Zhang. Physical media covert channels on smart mobile devices. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, page 367–378, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450335744. doi: 10.1145/2750858.2804253. URL <https://doi.org/10.1145/2750858.2804253>.
- Cesar Radar and Sandeep Rao. *The fundamentals of millimeter wave sensors The fundamentals of millimeter wave sensors 2*. 2017. URL https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/1023/fundamentalsmmwave.pdf.
- Nirupam Roy, Mahanth Gowda, and Romit Roy Choudhury. Ripple: Communicating through physical vibration. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 265–278, Oakland, CA, May 2015. USENIX Association. ISBN 978-1-931971-218. URL <https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/roy>.
- Roman Schlegel, Kehuan Zhang, Xiao-yong Zhou, Mehool Intwala, Apu Kapadia, and Xiaofeng Wang. Soundcomber: A stealthy and context-aware sound trojan for smartphones. 01 2011.
- Texas Instruments. *User's Guide for mmWaveStudio 4.1.0*. Texas Instruments Incorporated, Dallas, Texas, USA, 2022. URL https://dr-download.ti.com/software-development/ide-configuration-compiler-or-debugger/MD-6g8078B70G/04.01.00.01/mmwave_studio_user_guide.pdf. Copyright © 2022, Texas Instruments Incorporated.
-