UNIVERSITY OF OKLAHOMA GRADUATE COLLEGE

Understanding Privacy and Security Implications of Emerging ASR Devices

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the
Degree of

MASTER OF SCIENCE

By Yan Zhou

Norman, Oklahoma

2024

Understanding Privacy and Security Implications of Emerging ASR Devices

A THESIS APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY THE COMMITTEE CONSISTING OF

Dr. Song Fang, Chair

Dr. Anindya Maiti

Dr. Shangqing Zhao

© Copyright by Yan Zhou 2024
All Rights Reserved.

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my academic advisor, Professor Dr. Fang. During my study of Internet-of-Things (IoT) security, he gave me guidance in professional knowledge and valuable suggestions on research direction and thesis writing.

I would also like to thank my classmates in the laboratory. We help each other study and support each other in life, which makes me feel warm in the intense research and enables me to improve my professional ability. Especially when facing the test of life, their care makes me feel warm and encouraged.

In addition to academics, I would like to express my special thanks to the professors and advisors in my department and university. During the difficult period when faced with my mother's diagnosis of advanced lung cancer, their support allowed a focused and perseverant approach to studies.

Lastly, gratitude extends to everyone who provided support throughout the academic journey.

Table of Contents

ACKNOWLEDGEMENTS……………………………………………………………………………….iv

LIST OF FIGURES…………………………………………………………………………………vii

LIST OF TABLES…………………………………………………………………………………..ix

ABSTRACT…………………………………………………………………………………………x

CHAPTER I – INTRODUCTION TO REAL-WORLD ISSUES……………………………………….1

1.1 AN OVERVIEW OF ASR-EMBEDDED IOT DEVICES 1

1.2 PROBLEMS OF VOICE ASSISTANTS 3

1.3 A NEW TECHNIQUE FOR ASR DEVICE LOCALIZATION 6

CHAPTER II – LITERATURE REVIEW…………………………………………………………….9

CHAPTER III – SYSTEM MODEL…………………………………………………………………15

CHAPTER IV – ASR DEVICE LOCALIZATION…………………………………………………16

4.1 SYSTEM OVERVIEW 16

4.2 ASR IDENTIFICATION 17

4.2.1 Device Activation 18

4.2.2 Traffic Monitoring 20

4.2.3 MAC MATCH 21

4.3 Acoustic Signal Evaluation 23
4.3.1 Probability-Based Sound Collection 24
4.3.2 Incremental Window Accumulation 27
4.4 Distance Estimation 30
4.4.1 Difference Analysis 30
4.4.2 Position Bilateration 32
4.4.3 Localization Error 35
4.4.4 Special Cases 36
CHAPTER V – EXPERIMENTAL EVALUATION 38
5.1 EXPERIMENT SETUP 38
5.2 CASE STUDY 40
5.3 OVERALL PERFORMANCE 44
5.4 USER STUDY 47
CHAPTER VI – DISCUSSION 50
6.1 LIMITATIONS 50
6.2 FUTURE WORK 51
CHAPTER VI – CONCLUSION 53
REFERENCES 54

LIST OF FIGURES

Figure 1: Sound-activated ASR device, resulting in traffic change ………..…………………...…4

Figure 2: Overview of *SpyLoc* framework. …………………………...............................…16

Figure 3: Count of newly captured packets when a user activates an Amazon Alexa device..…………………………………………...…………………….………………........19

Figure 4: The unique traffic pattern observed during two consecutive activations of an Alexa ASR device compared to other non-ASR devices………..……………………………..............…..21

Figure 5: SVM training result for wireless traffic...23

Figure 6: Acoustic Model………………………… ……………..25

Figure 7: Distance vs. volume decibel required for ASR activation..26

Figure 8: Using the binary search to choose the volume level to wake up the ASR devices………………………...27

Figure 9: Incremental window size demonstration…..……………….......................................29

Figure 10: 2D Localization General Case……. ……………………..33

Figure 11: Special Case 1……………………………………………….........................…36

Figure 12: Special Case 2…………………………………………..…..………………........36

Figure 13: Experiment Environment Setup..……………………………..………………...…39

Figure 14: Localization error analysis in all 10 trials……………………..................................42

Figure 15: Localization time analysis in all 10 trials..42

Figure 16: Different environments localization error of Amazon Echo Dot................................44

Figure 17: Overall device localization error in environments A and B..46

Figure 18: Overall device localization time in environments A and B...46

Figure 19: Localization errors for all 10 users...48

LIST OF TABLES

Table 1: The list of ASR devices we test……………….....…..……………………………...…22

Table 2: Localization time for different devices by users…………………....…………….….49

ABSTRACT

The number of devices that use voice assistants has increased dramatically worldwide, reaching 4.2 billion in 2022 and is predicted to reach 8.4 billion by 2024. Voice assistants can interact with users through voice, perform specific tasks, and gradually learn and adapt to the user's habits. However, as voice assistants have grown in popularity, user privacy and security issues have become increasingly prominent. Topics such as misrecognition, potential eavesdropping, and waking up with hidden commands have attracted public attention. In this thesis, we propose an innovative method, *SpyLoc,* to locate widely used spying Automatic Speech Recognition (ASR) devices (e.g., Amazon Alexa, Apple Siri, and Google Assistant) to mitigate the potential surveillance risk. *SpyLoc* uses Text-to-Speech (TTS) to generate wake-up words and plays them at different positions with varying volumes to trigger the target hidden ASR. By analyzing the resultant wireless traffic generated by the ASR, we calculate the corresponding distances between the wake-up word player and the ASR. With spatial analysis, we can further pinpoint the location of the ASR. Our extensive real-world experiments using the developed application and three commercial off-the-shelf voice assistants show that *SpyLoc* can achieve low localization error with a short processing time (i.e., several minutes). This method presents an innovative approach to addressing the potential eavesdropping risks posed by ubiquitous voice assistants.

CHAPTER I

INTRODUCTION TO REAL-WORLD ISSUES

1.1 AN OVERVIEW OF ASR-EMBEDDED IOT DEVICES

Automatic Speech Recognition (ASR) devices have become pivotal in the smart device industry, transforming how users interact with technology. As industries evolve, businesses increasingly seek efficient ways to integrate smart technologies. According to Statista, approximately 4.2 billion devices will be equipped with voice assistants worldwide in 2022, projected to double to 8.4 billion by 2024—exceeding the global population [1]. In 2020, the United States reported over 142 million virtual assistant users, representing nearly half its population [2]. These trends suggest virtual assistants are poised to become the central hub for future home IoT (Internet-of-Things) ecosystems.

Virtual assistants have revolutionized how users interact with technology, offering personalized, voice-driven experiences that streamline daily routines. These devices can learn and adapt to user habits, enhancing convenience and efficiency. Imagine waking up in a room softly illuminated to simulate sunrise, eliminating the need to seach for a switch. As you prepare for the day, your virtual assistant operates the coffee maker, updates you on the weather, manages your schedule, and even places grocery orders. In a fast-paced world where time is invaluable, the ability to control every aspect of our living environments through simple voice commands has transformed how we engage with and experience life.

Natural language processing and speech recognition are core technologies driving virtual voice assistants. When a user inputs a voice command, such as "*Hey Google*" or "*Alexa,*" the device activates, and its microphones capture the sound, converting it into digital data. The signal is transferred to the cloud since IoT devices are not equipped to process and store all voice data [3, 4]. The algorithms in the cloud will interpret and parse all voice input and determine the most appropriate response. To minimize unnecessary data transfer. It is essential to note that these devices only record and transmit conversations when activated by a specific wake-up word to minimize unnecessary data transfer. Wake-up word detection is typically performed locally on the device to improve privacy and responsiveness. Once the device recognizes the wake-up word, it wakes up and responds to commands. The device returns to standby and waits for the subsequent prompt once the command is interpreted and executed. Data transmission occurs when the voice assistant is activated. This activation allows us to exploit this mechanism and correlate the device's behavior with the network traffic generated by the ASR device. Intuitively, we know that the traffic represents the presence of an ASR device.

The voice assistant market is experiencing rapid growth, driven by the convenience and functionality these devices bring to daily life. However, as people increasingly integrate these technologies into their homes, privacy and security concerns are becoming more prominent. A critical question arises: Could these smart home devices eavesdropping our conversations without our knowledge or consent? This thesis aims to address these concerns by exploring the feasibility of detecting and even pinpointing hidden Automatic Speech Recognition (ASR) devices. By leveraging proactive network traffic analysis, the study seeks to identify unique patterns associated with ASR device activity and correlate them with the location of the target ASRs. Our scheme offers a promising method to mitigate the surveillance risks posed by such devices, thereby enhancing individual privacy protection.

1.2 PROBLEMS OF VOICE ASSISTANTS

Voice assistants are most likely to function as surveillance devices among all the devices. First, they have multiple far-field microphones that can capture voice commands from anywhere in the room, even with background noise. Compared that to phones or smartwatches, which are built for close-range conversations. Second, they are always plugged in and powered, allowing them to listen for wake-up words like "*Alexa*" constantly. Finally, they are typically fixed in place in the home, often placed in central areas like the living room or kitchen.

Existing home IoT devices, such as smart speakers, voice assistants, and security cameras, often have built-in microphones for voice control and other functions. As of 2023, Amazon's Alexa is integrated into nearly 1 billion devices globally, significantly rising from around 60,000 smart home devices in 2019, indicating the software's widespread adoption [5, 6].

Misrecognition is a significant issue with voice assistants' speech recognition technology (see Figure 1). On the left of the figure, the user stays silent, but data fluctuates because other devices communicate via wireless signals, and the transmitted data is captured. However, voice assistants are inadvertently activated on the right by mishearing other words as wake commands from the user. The issue of misrecognition rates cannot be fully resolved by deep learning and other algorithmic models as they are fundamentally based on probability [7, 8]. *Mitev et al*. introduced the *LeakyPick* architecture and identified 89 words that the Amazon Echo Dot misinterprets as its wake-up word [9]. This implies that smart devices can be unintentionally activated by misrecognizing other words as a wake command. Once the voice assistant is activated, it begins to record and transmit the audio data to the cloud for processing, which could result in the user's conversation being monitored.

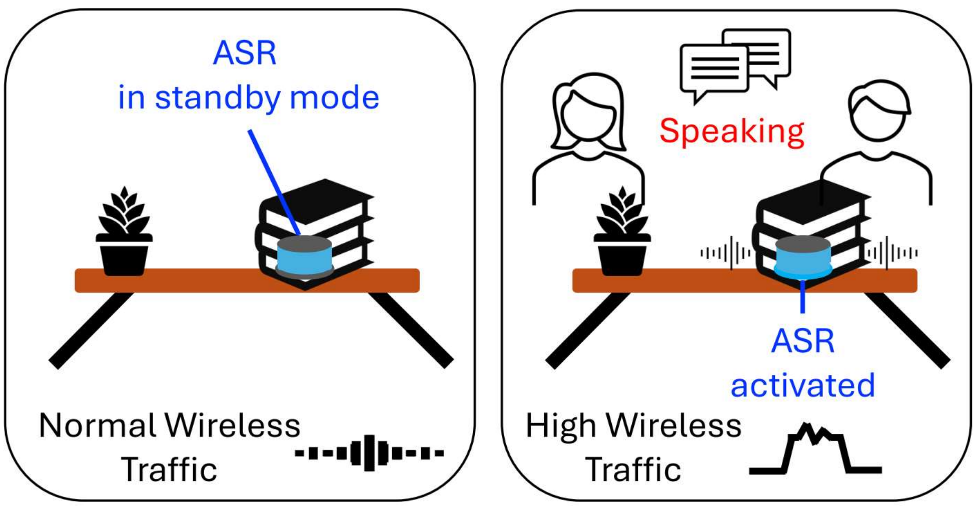

Figure 1: Sound-activated ASR device, resulting in traffic changes.

Furthermore, some people might utilize voice assistants as a hidden surveillance tool. For instance, the ASR devices could be placed near the user without the user's permission and activated with hidden commands while disguising itself as another commonplace item, such as an electronic clock or air freshener [10, 11, 12]. *Zhang et al*. proposed DolphinAttack, a type of hidden command [13]. It is completely inaudible and unrecognized to humans, but it is recognized by the

ASR device through modulation of voice commands on ultrasonic carriers. Hidden surveillance especially happens in intimate relationships; monitoring a partner's or other family member's activities and privacy may be abused, known as intimate partner surveillance (IPS) [13].

In April 2019, Amazon employed thousands of foreign contractors to listen to and analyze user conversations with its Echo smart speakers, according to reports from Bloomberg, CBS, CNN, and numerous other media outlets [14, 15, 16]. Amazon claimed that the company only uses a minimal number of user conversations with voice assistants in order to enhance its natural language understanding systems and thereby further enhance the user experience. However, Amazon does not explicitly tell consumers that its employees are listening in on their conversations with Alexa. Following this, Apple and Google subsequently disclosed instances of employing workers to listen [17, 18]. The instances sparked widespread controversy over privacy and data use, raising public awareness of privacy and security issues related to smart device use.

As of right now, precise data regarding the quantity and seriousness of occurrences utilizing hidden surveillance devices are unavailable. Some professional tools have been developed to detect radio frequency, infrared radiation, and other unusual signals that indicate hidden devices. However, there are still challenges. Physical searches are time-consuming and rely heavily on human effort. Infrared scanning can be thrown off by ambient temperature or external light [19]. Radio frequency scanning can pick up signals but cannot tell you exactly where the device is. Accurately locating hidden surveillance devices is still challenging due to these considerations [13].

Even in the latest research, similar IoT devices have no accurate localization. *Zhou et al*. detected an operating hidden recorder by capturing electromagnetic radiation (EMR) emitted into the environment. However, their detection range was limited to 0.5 meters, achieving an overall success rate of 92.17 percent at a distance of 0.2 meters [20]. Even in the past decade, WiFi trilateration or fingerprint identification techniques, considered to have the greatest potential for indoor localization and navigation, cannot accurately locate ASR devices [21]. *SpotFi* can only achieve an accuracy of 40cm at most [22]. *AoA* localization also has a maximum error of (0.24) meters for the X coordinate and (0.20) meters for the Y coordinate and requires 30 Receivers (RXs) and 2 Transmitters (TXs) placed in appropriate locations [23]. The above evidence shows that existing detection methods are unsuitable for locating hidden ASR devices. There is still a gap in the research on the precise location of hidden ASR devices such as the Amazon Echo Dot.

1.3 A NEW TECHNIQUE FOR ASR DEVICE LOCALIZATION

In this thesis, we proposed *SpyLoc*, mainly to localize ASR devices. We use Google Cloud Text-To-Speech (TTS) to generate the wake-up word for the voice assistant and play it at two fixed locations with different volumes to test the distance between the voice and voice assistant locations. Initially, we tried to find the lowest volume that wakes the ASR device at a specific distance and created a mapping to determine the distance corresponding to the lowest volume in testing. However, the ASR device listens for the wake-up word through a small, continuously running speech recognizer that uses a deep neural network to convert the audio signal into a probability distribution of speech categories, combined with a hidden Markov model and a recurrent network to track the timing of the speech signal. Features and accumulated speech category scores determine whether to activate Siri [24]. To put it simply, the threshold that determines whether to activate the ASR device is not a fixed value. So here, we will play the wake-up word at a distance multiple times and record its wake probability. The calculation uses the wake-up probability, and the calculated data is saved to help find the distance. Next, determine the voice assistant's location by calculating two distances.

However, since various devices (such as smartphones, computers, smart home devices, etc.) communicate via wireless signals in the wireless network environment, there is a significant amount of data transmission in the air. However, every network device is uniquely identified by its permanent MAC address. The manufacturer or supplier's Organizationally Unique Identifier (OUI) is shown in the first half, and the unique device ID is represented in the second half [25]. We can observe that all network packets are transmitted over the air, and we can identify the specific MAC Address of every ASR device. Identifying a particular type of device based only on the MAC address may need additional work because some manufacturers create many network equipment types. Therefore, we must combine the MAC address and network activity to determine whether it is an ASR device. The voice assistant is then checked to see if it has woken up by tracking the number of new data packets received every second.

Our extensive experiments using the developed application and three popular ASR devices show that *SpyLoc* can achieve a mean localization error of about 4.94 inches in an average of less than 300 seconds. This method provides an innovative solution to the potential eavesdropping risk of ASR devices.

The main contributions of this work are summarized as follows:

• This study proposes the first practical method to accurately localize hidden ASR devices, which may stealthily record sensitive conversations.

• We develop a novel approach for correlating ASR activation with wireless traffic patterns.

• We evaluate the effectiveness and efficiency of the proposed ASR localization technique.

• We implement SpyLoc and consistently achieve high localization accuracy with an average of less than 300 seconds.

CHAPTER II

LITERATURE REVIEW

The rapid development of big data and artificial intelligence (AI) has become one of the most prominent research topics in science and technology, with privacy and security issues drawing significant attention. Big data is essential for the training, identification, and prediction processes of AI algorithms, enabling them to deliver increasingly precise and intelligent services. Encouraging people to permit Internet companies to use their data explicitly is one of the most effective strategies to defend their right to know and choose. Sharing personal information should not equate to granting internet companies unrestricted access to sensitive data. One effective way to protect individuals' rights is to encourage explicit consent for data usage, ensuring transparency and choice. Companies like Apple have set a precedent by clearly stating in their policies that they can only borrow user information for specific purposes and are prohibited from sharing it with third parties [26].

Anti-spying Techniques: Conversations recorded by ASR devices are inevitably transmitted over the Internet and stored in the cloud. However, unauthorized access can result in users being under unknown surveillance. To prevent hackers or other organizations from stealing users' private information, manufacturers have taken a variety of actions:

- *Data Encryption*: Encrypt the information between the device and the cloud server. Strong encryption technology ensures that even if the data is intercepted during transmission, it cannot be deciphered without the correct decryption key [52]. This maintains the confidentiality of data exchanged between the user and the device, reducing the risk of unauthorized access.

- *Physical Switch or Mute Button*: This feature, which enables users to turn off the camera or microphone when not in use, is present in many smart devices [27]. This allows users to manage when their device is actively listening or recording proactively.
- *Transparent Privacy Settings*: The manufacturer has introduced more transparent privacy settings. Users can view and modify the permissions granted to their devices and select the data they are willing to share [28]. This transparency lets individuals customize the device's functionality to match their privacy preferences.

Acoustic-based Defenses: In 2021, researchers at Columbia University developed an innovative predictive attack defense method to mitigate the surveillance risks of smart devices. In order to prevent smart devices from accurately transcribing spoken language, judging the user's subsequent conversation generates unique background noise in real-time [29, 30]. However, the technology currently works only in English, with a success rate of about 80%, and it is difficult for machines to keep up with human speech. Another system called *InfoMasker* system, introduced in 2023, leverages an interference noise into a nearby microphone so that even if the device records the sound, it is unable to extract any information about the user [31]. *InfoMasker* method safeguards against eavesdropping while ensuring users remain unaffected by noise and prevents the injected noise from being removed by denoising mechanisms. This system effectively reduces the accuracy of speech recognition systems to less than 50 percent, even when the Signal-to-Noise Ratio (SNR) is 0.

Wireless Traffic Analysis: Wireless traffic analysis refers to examining and interpreting patterns in network communication to gain insights into the behavior of wireless devices, systems, or users without necessarily accessing the actual content of the traffic. Due to the openness of wireless media, wireless signals can be intercepted and analyzed to infer sensitive information [32]. Traffic analysis is a standard technique used in many domains, including improving application performance [33], detecting anomalies and vulnerabilities [34, 35], and capturing user information on web applications [36]. In an era where IoT is widely used, many researchers and organizations utilize traffic analysis techniques to analyze the behavior of IoT devices and their users. Wireless traffic analysis with machine learning techniques effectively characterizes user behavior on IoT devices, classifies encrypted IoT traffic, and reveals user activities from IoT device traffic data [37, 38]. MotionCompass proposed an innovative approach to use wireless traffic analysis to identify traffic from wireless cameras and analyze traffic patterns to localize hidden cameras [25, 39]. In Automatic Speech Recognition (ASR), many recent studies have shown that it is feasible to classify the voice commands sent by voice assistants from network traffic with a high accuracy of 93.36% [38, 40]. Unlike previous works, which utilized traffic analysis to detect user activity and behavior for ASR, our work correlates traffic patterns with wake-up possibilities to infer the location of ASR devices.

ASR devices often rely on specific wake-up words to activate, interact with users, and execute their commands. Many people believe there should be a distinct decibel threshold for the wake-up word to activate the ASR device, and when the user volume is above the threshold, the ASR device will be active [41]. However, the wake-up word does not always successfully activate the ASR device 100 percent, even if the user increases the volume and tries repeatedly, which has led to in-depth research on the trigger probability of ASR devices [42, 43].

Wake-Up Word Detection Mechanism: In standby mode, ASR devices continuously listen to the sounds in the surrounding environment through built-in microphones [4]. The device initially uses a low-power neural network to identify the wake-up word to save energy and computing resources. This lightweight system detects features related to the wake-up word in the audio instead of processing the full voice command. When the user says a wake-up word such as *"Siri"* or *"Alexa,"* the audio signal is captured by the microphone and analyzed in the frequency domain to extract features such as Mel Frequency Cepstral Coefficients (MFCC); these features help the system determine whether the audio signal may contain the wake-up word [44].

Process Audio and Generate Confidence Scores: The system uses RNN (recurrent neural network) to process the time series information of the audio signal [45]. Since the wake-up word is a phrase composed of multiple phonemes, RNN can process the audio frame by frame, remember the contextual information of the previous and next frames, and combine the information of the current frame. The advantage of RNN is that it can capture the temporal dependency of speech, gradually accumulate the scores of each frame, and generate a total confidence score, which indicates the degree of match between the current audio signal and the wake-up word.

In this process, HMM (hidden Markov model) is used together to model the time series characteristics of audio [2, 46]. HMM regards the speech signal as a series of state transitions; each state corresponds to a part of the audio feature, and these state transition models are used to determine whether the wake-up word pattern appears. However, with the development of deep learning technology, RNN has gradually replaced HMM, especially RNN variants such as LSTM and GRU, which can better handle long-term dependencies and avoid the gradient disappearance problem.

In the last step of wake-up word detection, the RNN model generates a total score that matches the wake-up word based on the confidence scores accumulated frame by frame. This score is compared with the threshold in the system. If the score exceeds the set threshold, the system will determine that the user has said the wake-up word and the device will be officially activated. If the score is lower than the threshold, the system will ignore this audio clip and maintain a low power state, waiting for the next wake-up word to appear.

Multi-Threshold Activation and Misidentification Risk: The Apple development team mentioned that the threshold is not a fixed value [24]. The system uses two thresholds—a primary and a lesser threshold—to facilitate the activation of Siri under challenging conditions. If the score is in between, the system enters a more sensitive state for a short period to catch possible Siri events. This second chance mechanism significantly improves system availability while controlling the false alarm rate.

Although deep learning and other algorithm models have greatly improved the recognition accuracy of voice assistants, they cannot completely eliminate misrecognition because these models are essentially based on probability. This means that voice assistants may be accidentally activated [7,8] because they mistakenly recognize other words as wake-up words. The LeakyPick architecture proposed by Mitev et al. shows that Amazon Echo Dot can misidentify 89 different words as wake-up words [9]. Once mistakenly activated, the voice assistant will start recording and transmitting the audio data to the cloud for processing, which may cause the user's conversation to be recorded or eavesdropped without knowing it.

CHAPTER III

SYSTEM MODEL

We consider a general scenario where an attacker secretly deploys an ASR device in an area, such as behind a television, covered by a cloth, or hidden inside flower pots or books. The attacker aims to stealthily record potential conversations, posing a significant privacy threat. To counter this, *SpyLoc* is designed to pinpoint the location of such hidden ASR devices.

SpyLoc operates by playing wake-up words at varying volumes from a smartphone at different locations within the area. These wake-up words are intended to trigger the hidden ASR device, which we assume can be activated by a pre-recorded or synthesized voice [47]. Also, we assume that *SpyLoc* can sniff the wireless network traffic generated by the ASR device and is capable of moving the smartphone to different positions to enhance detection accuracy.

Besides, we consider scenarios where the ASR devices do not produce any audible feedback upon activation, either by default or because their audio response functionality has been manually disabled. In these cases, the ASR device still processes the wake-up word and generates corresponding network traffic. *SpyLoc* leverages this behavior by correlating the ASR device's activity with the observed traffic patterns, enabling effective detection and localization even when the device remains silent after activation.

CHAPTER IV

ASR DEVICE LOCALIZATION

4.1 SYSTEM OVERVIEW

SpyLoc focuses on two key areas: database building and ASR device location, as shown in Figure 2. To create the database, we go through phases one and two (ASR identification and acoustic signal evaluation). For device location, we first go through phases one and two; then, we get into phase three (distance estimation).

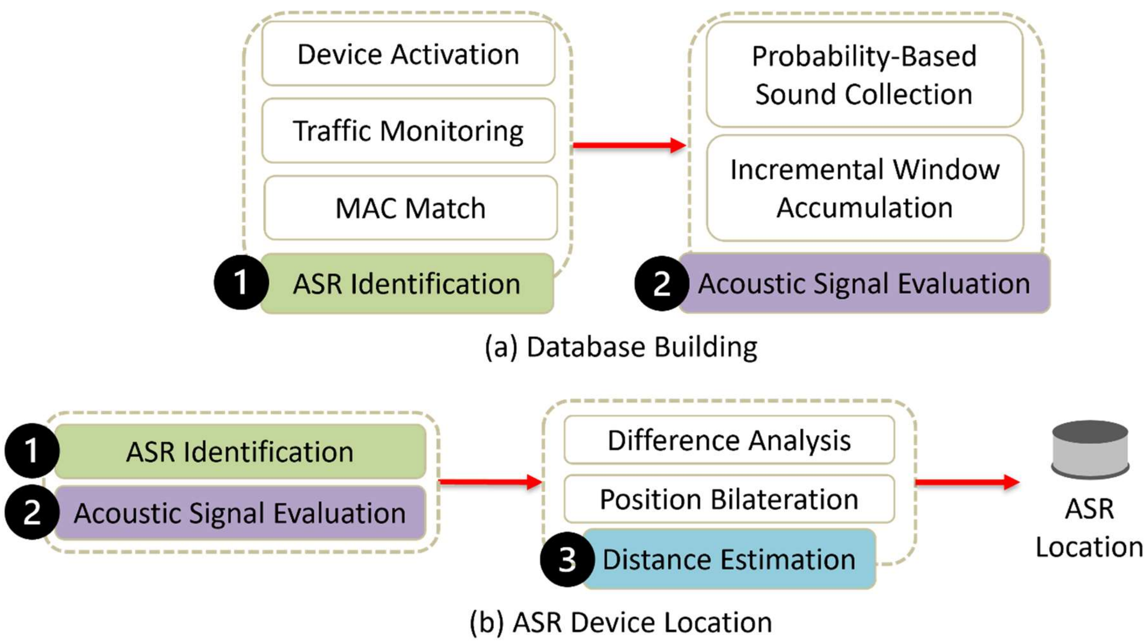

Figure 2: Overview of *SpyLoc* framework.

The first stage determines whether the user is subject to hidden ASR device monitoring. The user plays the wake-up word in the area of interest. If the number of WiFi traffic packets increases significantly during this period, a hidden ASR device may be listening in the area. We collect the MAC address of each traffic and compare it with the OUI part of the MAC address of a known ASR device to find possible hidden voice assistant devices and monitor all traffic related to that MAC address. The second phase will collect and process the data by accumulating it. The user randomly selects two locations in the area, plays the wake-up word at different volumes, and records the wake-up rate. In order to better distinguish the changes in wake probability at different distances, we perform accumulative processing. Each distance is accumulated multiple times to better distinguish different distances. The third stage will conduct differential analysis by comparing actual and experimental data to determine the straight-line distance between the user's randomly selected position and the ASR device. Then, we take the user's actual position as the center of the circle, the straight-line distance from the ASR device as the radius, and we use the intersection of the circles to calculate the exact position of the ASR device. We will present the details of each stage in the discussion below.

4.2 ASR IDENTIFICATION

Since various devices (such as smartphones, computers, smart home devices, etc.) communicate through wireless signals in a wireless network environment, there is a large amount of data transmission in the air. We need to find and monitor the traffic of hidden ASR devices to identify if users are being spied on. We recommend playing the wake-up word in the target area to stimulate the ASR devices to listen and monitor all traffic to determine whether any traffic produces a significant amount of new packets and becomes a candidate for hiding the ASR device.

4.2.1 Device Activation

Methods of activating Automatic Speech Recognition (ASR) systems are evolving beyond traditional button pressing. These include using the button on the headset [48], vibration through rapid shaking [33], and using gesture recognition technology to enable users to activate ASR systems via predefined actions captured by cameras or sensors [50, 51]. However, the most common one is to use the corresponding wake-up word to activate the ASR device.

Natural language processing and speech recognition are the fundamental ideas of voice assistants. Wake-up-word speech recognition enables ASR devices to remain in a low-power standby state and only initiate standard recognition when a wake-up word is detected, such as *"Hey Google"* or *"Alexa"* [52]. Since IoT devices often lack enough computing power and storage space to process all the voice data, once the wake-up word is recognized, the signal is usually sent to the cloud for further processing [3,4]. Advanced algorithms parse the speech content in the cloud and determine the most appropriate response. These algorithms are capable of not only recognizing spoken language but also understanding the meaning of the language. Once the command is interpreted and executed, the device returns to standby, waiting for the following prompt.

The data will be transmitted through routers in the network and generate wireless traffic as it is sent to the cloud. During this procedure, the user can play the wake-up word of the voice assistant in the target area to stimulate the ASR device to listen in the area. The voice assistant will be activated, and the content will be recorded following the wake-up word. As shown in Figure 3, when the ASR device (Amazon Echo Dot) is in the standby state, a minimal amount of new packets will be generated every second, almost approaching 0. If a wake-up word is detected, high wireless traffic will be generated. In the activated state, the packets generated per second gradually increase until the voice assistant re-enters standby mode.

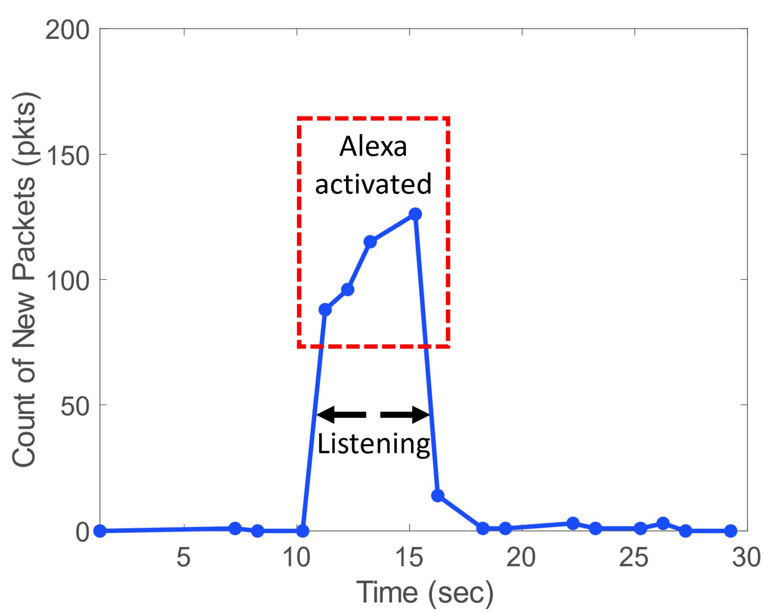

Figure 3: Count of newly captured packets when a user activates an Amazon Alexa device.

4.2.2 Traffic Monitoring

ASR devices typically feature low-power designs to extend battery life and reduce energy consumption. As a result, the device will reduce cloud communication while in standby mode, lowering traffic generation [53]. Once the wake-up word is detected and the speech is recognized in text form, this text data is typically sent to the cloud for further processing and analysis, generating an appropriate response [3]. During this period, the traffic generated by the ASR device shows a distinguishable pattern; that is, the amount of traffic generated by the ASR device depends on whether the ASR device is activated.

Here, we use *aircrack-ng*, a suite specifically designed for wireless network monitoring and analysis on Kali Linux. Users can set a wireless device into monitor mode, allowing the network adapter to capture and analyze all packets traveling over the air, including communications between devices connected to the network (see Figure 4). Not all network adapters or NICs support monitor mode; we chose the Alfa AWUS036NHA here. If monitored wireless traffic suddenly increases significantly when the wake-up word is broadcast, that traffic can be identified as a candidate for an ASR device, and the MAC address is recorded.

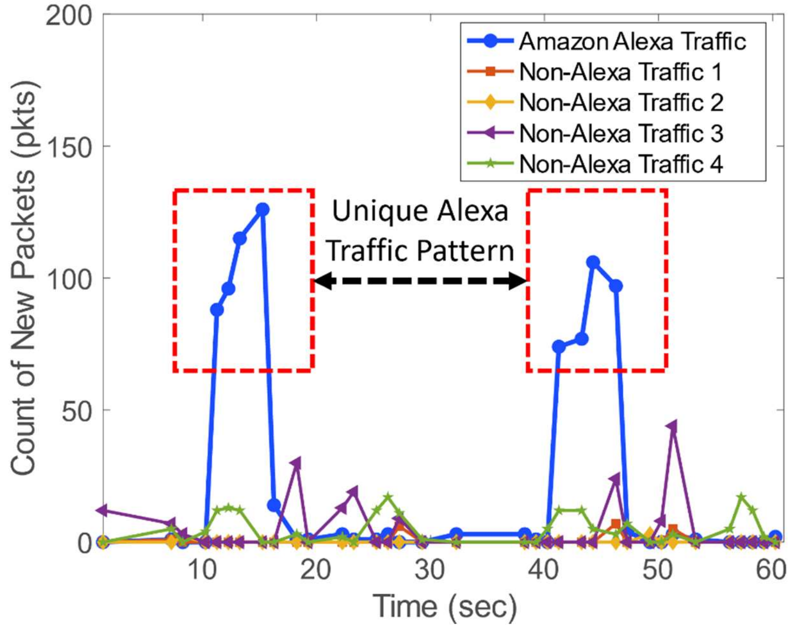

Figure 4: The unique traffic pattern observed during two consecutive activations of an Alexa ASR device compared to other non-ASR devices.

4.2.3 MAC MATCH

Each network device has a unique and permanent MAC address for identification at the physical network level. A MAC address usually consists of six groups of two hexadecimal digits separated by colons or hyphens. The first half represents the manufacturer or supplier's OUI (Organizationally Unique Identifier), and the second half represents the unique device ID. The uniqueness of OUI prompted us to build one for all common ASR devices on the market, as shown in Table 1. We can determine whether the device's manufacturer belongs to the device we detect by comparing the prefix of the MAC of the candidate ASR device with the publicly available prefix defined by the manufacturer [54].

Table 1: The list of ASR devices we test.

Device Name	ASR Model	Wake Word
Echo Dot 5th Gen	Amazon Alexa	*Alexa*
Nest Mini 2nd Gen	Google Assistant	*Hey Google* or *OK Google*
HomePod Mini	Apple Siri	*Hey Siri* or *Siri*

MAC Spoofing: Though MAC addresses can be used for identification at the physical network level, a device can broadcast to broadcast a MAC address that is different from the OUI-based MAC that the manufacturers assign [25]. Some devices allow users to modify the MAC address arbitrarily through software [55, 56]. Therefore, the user could bypass the ASR device traffic detection by using an MAC address that is not associated with any ASR device manufacturers or adding noise to the traffic detection process by assigning a non-ASR device with a MAC address of an ASR device. To resolve the problem, there are recent studies that proposed a technique using a unique identifier called the Universally Unique Identifier-Enrollee (UUID-E), which can effectively recover a device's original global MAC address, even if it has been spoofed or randomized [55, 57, 58]. The proposed OUI-based traffic analysis remains effective by integrating these methods to restore the actual MAC addressing cases of MAC spoofing.

Since the traffic generated by the ASR device shows a distinguishable pattern, we can use the pattern to propose another solution to address MAC spoofing. The distinguishable traffic pattern of ASR devices motivates us to train a Support Vector Machine (SVM) model to classify traffic patterns, which can then be used to identify whether a given traffic instance is from an ASR device.

The SVM classifier model is built using the Scikit-learn library. We grouped traffic data into smaller sets of traffic in a given time frame and extracted features of the average traffic, the maximum number of traffic, the minimum number of traffic, the standard deviation of the traffic, and the total traffic of the device in the given time frame to train the SVM model. These features were used to distinguish ASR device traffic from non-ASR device traffic. Since our training data contains more than two features, we applied Principal Component Analysis (PCA) to reduce the feature space to two dimensions for better visualization. Figure 5 shows the result of running SVM on 6400 traffic flows from both ASR and non-ASR devices, demonstrating the effectiveness of distinguishing traffic flows from the ASR devices.

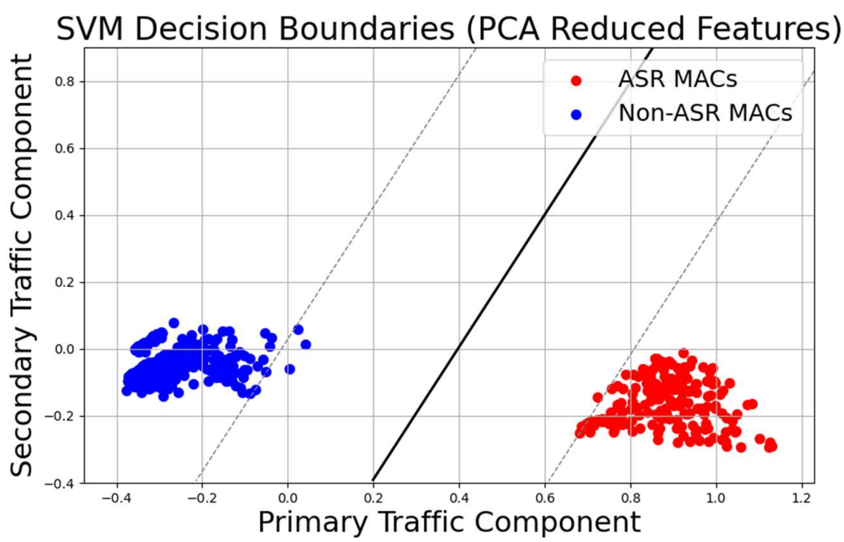

Figure 5: SVM training result for wireless traffic.

4.3 Acoustic Signal Evaluation

ASR typically uses a Keyword Recognition System (KWS) running locally on the terminal device to detect wake-up words and then uses a Hidden Markov Model (HMM) to model the temporal process of the speech signal, where each state corresponds to a possible Voice category. Then, the speech category score of each frame is accumulated through temporal integration and recurrent network to detect whether the user has spoken the wake-up word [59, 24]. Since the probability of awakening is not 100 percent, even if you increase the volume and try multiple times. Therefore, the user collects the probability of waking up the ASR device at different volumes at two randomly selected locations within the target area.

4.3.1 Probability-Based Sound Collection

We observed that when ASR devices are placed at a farther distance, a higher sound pressure level is required to activate them due to the reduction in sound pressure level as they propagate over distance. As discussed in the Springer Handbook of Acoustics [60], the reduction in sound pressure level per distance can be described in the following equation:

$$SPL_2 = SPL_1 - 20 \times \left(\frac{d_2}{d_1}\right) \tag{1}$$

By setting d_2 to 1, we can use this equation to find the sound pressure level required to find an activated voice assistant at a given distance from the sound source:

$$SPL_a = SPL_i + 20 \times (d_2) \tag{2}$$

Here, SPL_a is the sound pressure level required to activate the ASR, SPL_i is the initial sound pressure level required to activate the ASR where the sound source is precisely 1 inch away, and d_2 is the distance from the sound source where SPL_i is measured. As shown in Figure 6, we can see that our data follows the overall equation pattern with some variance since each voice assistant has its unique algorithm to detect wake-up words. In Figure 7, we demonstrated the required volume level to play the wake-up word using our testing device to activate ASR devices at a given distance. We can utilize this property and the minimal volume level needed to activate the ASR to approximate the current distance of the ASR device from the current location where the sound was played. However, due to the impact of noise, the minimal volume cannot guarantee device activation every time; thus, we will use wake-up probability as our primary metric to determine the minimal volume needed.

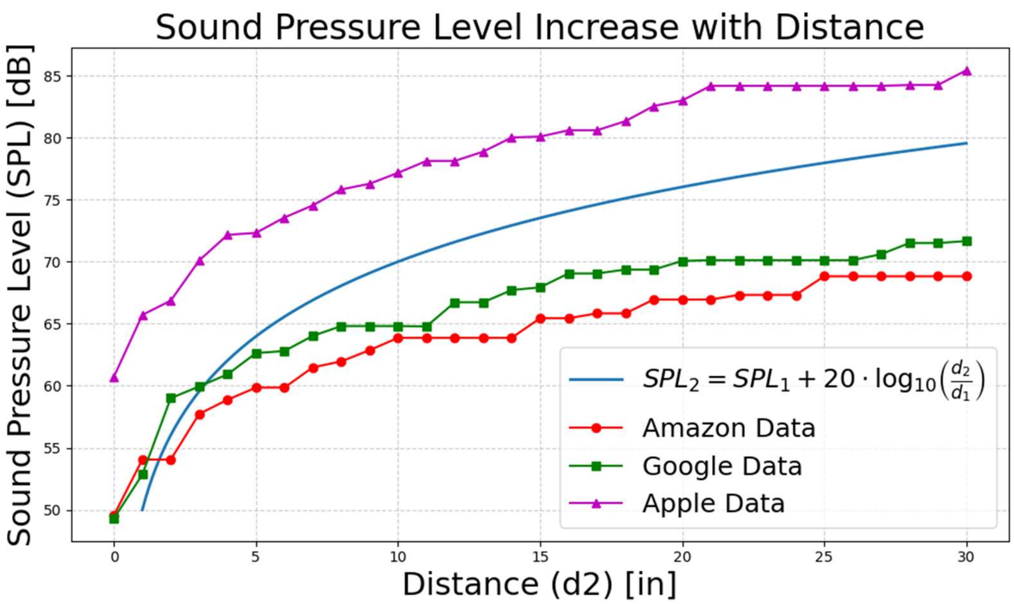

Figure 6: Acoustic Model

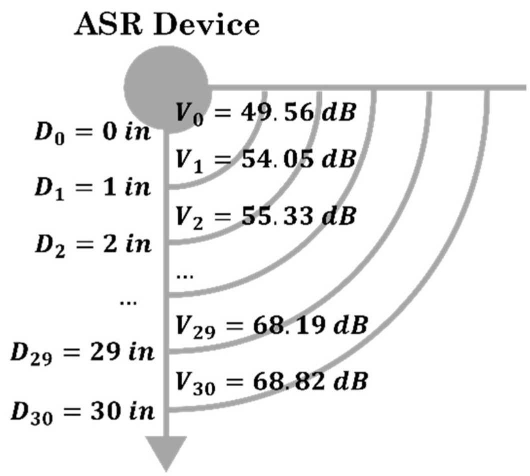

Figure 7: Distance vs. volume decibel required for ASR activation.

Human sound production is a complex process involving the coordinated movement of multiple organs, such as the vocal cords, throat, oral cavity, and nasal cavity. Although we can adjust the volume of the sound by controlling our breathing and muscles, this adjustment is often relatively crude and does not consistently achieve the precise target [61, 62]. So, we used Google Cloud TTS technology to generate ASR wake-up words and used Samsung A50 mobile phone to play them at different volumes to ensure the consistency of the voice and intonation for each measurement.

To save localization time, we use binary search for more efficient localization and search. We first play the wake-up word at a specific volume at any location in the suspected area. Suppose the monitored wireless traffic does not respond when the wake-up word is played. In that case, the distance of the ASR device from the location is beyond the inspection range. We will repeat the detection outside the area with a radius of 30 inches and the location as the center point until the

ASR device is successfully awakened. The approximate distance between the detection and ASR devices is determined based on the successful awakening rate. Next, we can use binary search to select the appropriate playback volume and implement more accurate localization. We need to find the last volume that cannot wake up the ASR and record the following three wake-up probabilities. We first move to the middle between zero and the highest volume recorded by the device and then perform a wake-up test on this volume. If this volume wakes up at least once, we need to use binary search to reduce the volume. If this volume cannot wake the ASR, we repeat the binary search to increase the volume (see Figure 8). Until only the last data is left, it is the point we are looking for.

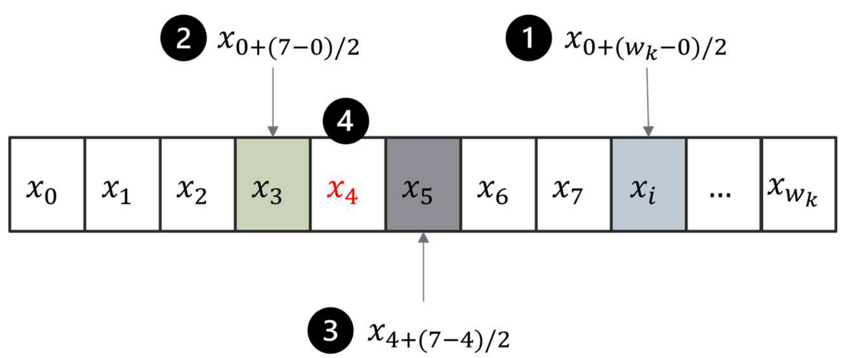

Figure 8: Using the binary search to choose the volume level to wake up the ASR devices.

4.3.2 Incremental Window Accumulation

Any signal is inevitably interfered with by the external environment. In particular, when sound is propagating, due to the reflection of sound, part of the sound signal is inevitably lost each time, resulting in different sound signals received by the ASR device each time [60]. To address the issue of signals being challenging to capture due to noise, Tang et al. proposed a type of accumulation sliding window algorithm in the article to improve the signal detection capability [63]. In order to delve deeper into the specific impact of volume level on the probability of ASR activation under different distance conditions, we used an additive processing method to analyze the data. When the distance is close, the same volume may have similar or even the same probability. However, when we accumulate data in a window, these similar differences will be added to better distinguish different distances. This method involves summing each column of data (i.e., different volume levels at the same distance). This way, we can calculate the overall probability of activating the ASR device for different volume combinations at each distance.

Data accumulation processing: For a series of volume levels of data $x_0, x_1, \ldots, x_{w_k}$ at a specific distance, we can use the following formula to calculate the sum of all data point values within the window:

$$\sum_{i=0}^{W_k} x_i \tag{3}$$

Here, w_k represents the window size currently considered and x_i represents the number or probability of activating the ASR at the i-th volume level within the window. By calculating this sum, we are able to get the total number of times or the total probability that the ASR was activated for all volume levels considered.

Stepwise increase in window size: As shown in Figure 9, we start with an initial window size w_0 of 5. Subsequently, the size of the window will gradually increase to include more volume levels and the increasing process can be expressed by the following formula:

$$w_{k+1} = w_k + 1 \tag{4}$$

Until the window size w_k equals the total length of the data sequence n, k is the increasing step's index.

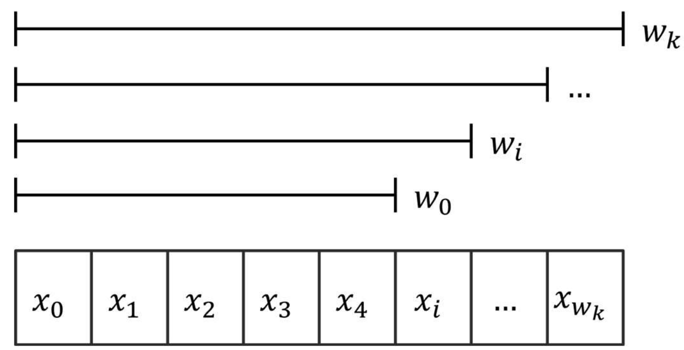

Figure 9: Incremental window size demonstration

Through this method of cumulative processing and gradually increasing the window, we can comprehensively consider the impact of different volume levels on the ASR's activation probability. We have enhanced the stability of the data and reduced the impact of single abnormal data on the overall results. This allows us to evaluate the impact of individual volume levels and observe trends in the overall activation probability as more volume levels are included.

4.4 Distance Estimation

After processing the data, we need to explore the relationship between the ASR device activation probability and distance and volume level and achieve positioning. First, through the difference analysis method, based on the difference of probability data at different distances and volume levels, the difference between the data collected at random locations and the overall data set is calculated. Then, the absolute minimum value of each data set is tracked to gain critical insights. Subsequently, the distance from the random location to the ASR device is estimated based on the difference value data, and then the possible location of the ASR device is determined using this distance information. In determining the position, the uncertainty that may result from actual measurement errors is considered, and the intersection problem between circles is solved by calculating the intersection point. At the same time, targeted solutions are proposed for two special situations to ensure the system model can effectively handle abnormal situations.

4.4.1 Difference Analysis

In order to explore the impact of different distances and volume levels on the activation probability of voice assistants, we adopted a method based on dissimilarity analysis. The method's core is calculating the difference between probabilistic data collected at randomly selected locations within the target area and the overall data set. Specifically, we use the following formula to calculate dissimilarity:

$$D_{ik} = M_{ik} - M'_k \tag{5}$$

In this formula, D represents the difference between the two data sets. M is the data after accumulating all volume levels at different distances, and M′ is the data after accumulating all volume levels at one distance. i represents the row index, and k represents the column index. If multiple difference values in a window are close to zero, this indicates that the representation of this window in the two data tables is very similar (it may also represent an invalid feature).

Further, we track the absolute minimum value of each set of data in the random position difference value data. These absolute minimums may provide key insights about specific distances and volume levels. In some cases, there are many zeros in the data, which may indicate that the features of this window do not provide meaningful information (the data in the two tables are almost identical). This may occur when the distance between the wake-up device and the ASR device is large. Since the low decibels are insufficient to trigger the ASR device to wake up, neither data set has an effective response under this condition. It resulted in a large number of zero values in the data, so we chose to exclude it. Removing these meaningless windows retains more discriminative windows, making the final results more reliable and representative. Then, we use the following formula to find the row index corresponding to the absolute minimum value of each column of data:

$$i_{min\,(k)} = arg|D_{ik}| \tag{6}$$

In this formula, $i_{min\,(k)}$ is the row index corresponding to the absolute minimum value of column k. These row indices can find the distance corresponds to the absolute minimum value. The minimum value of the distance can be interpreted as the closest position of the two sets of data within this window size, potentially reflecting the value where the actual data is most similar to the experimental data. We then calculate the average of the row indices corresponding to the absolute minimum value of all columns, which provides a representative distance of overall data to obtain the estimated distance from a random location to the ASR device, as follows:

$$r = mean\ (i_{min}) = \frac{1}{n} \sum_{k=0}^{n} i_{min\ (k)} \tag{7}$$

Among them, n is the number of columns, i_{min} is the set of $i_{min\ (k)}$ of all columns. This average estimates the distance to the ASR device from randomly selected locations within the target area.

4.4.2 Position Bilateration

After we confirm the distance to the ASR device from a randomly selected location in the target area, we can use this information to calculate its location. First, we generate a circle using the coordinates of a random location as the center and the distance from the random location to the ASR device as the radius. We then find the intersection points of these circles, which are the possible locations of the ASR device (see Figure 10).

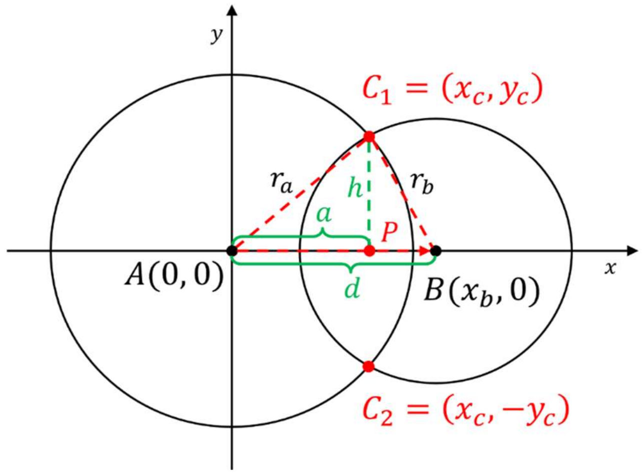

Figure 10: 2D Localization General Case

In order to accurately determine the location of the ASR device, we first calculated the straight-line distance d between two random locations, whose formula is as follows:

$$d = \sqrt{(h_b - h_a)^2 + (k_b - k_a)^2} \tag{8}$$

Among them, (h_i, k_i) are the coordinates of two random positions, and the indices a and b represent the first and second random positions, respectively. This gives us the straight line distance from the center of the circle, which is the straight line distance between random locations. Next, we calculate two temporary variables, a and h. As marked in Figure 7, the variable a is the distance between the midpoints of the line segments from the center of a circle to the two intersection points of two circles (if they intersect). h represents the distance between the midpoint of the intersection line and the intersection of the two circles. Their calculation method is as follows:

$$a = \frac{r_a^2 - r_b^2 + d^2}{2 * d} \tag{9}$$

$$h = \sqrt{r^2 - a^2} \tag{10}$$

Among them, r_a and r_b are the radii of the two circles, respectively. Next, we find the point P on the line connecting the centers of the two circles. It is located on the line connecting the centers of the two circles. It is also located on the line connecting the intersection points of the circles. We calculate the x and y coordinates of point P by the following formula:

$$P_x = \frac{h_a + a * (h_b - h_a)}{d} \tag{11}$$

$$P_y = \frac{k_a + a *(k_b - k_a)}{d} \tag{12}$$

Finally, we use the coordinates of point P and the vertical distance h to calculate the coordinates of the intersection of the two circles. Typically, there will be one or two possible solutions corresponding to the two intersection points:

$$C_1 = (P_x + \frac{h * (k_b - k_a)}{d}, P_y - \frac{(h * (h_b - h_a))}{d}) \tag{13}$$

$$C_2 = (P_x + \frac{h * (k_b - k_a)}{d}, P_y - \frac{(h * (h_b - h_a))}{d}) \tag{14}$$

By finding the intersection of these two circles, we can determine the likely location of the ASR device. If there are two intersection points, we will choose the most appropriate one as the location of the ASR device based on the actual situation.

4.4.3 Localization Error

In the above system overview, we assumed an ideal situation and used the centers of different circles and their radii as input parameters to find their intersection points. This can be expressed as the following function:

$$C = f\ (h_a, k_a, r_a, h_b, k_b, r_b) \tag{15}$$

where a represents the data of the first random point, and b represents the data of the second random point. However, in reality, it is not easy to achieve perfection. Measurement errors may occur during the experiment, leading to uncertainties in different radii and intersection points. Therefore, we must consider the actual radius r′ and add the error Δ, that is:

$$r = r' + \Delta \tag{16}$$

Considering possible errors, in order to find the exact intersection point, we need to calculate the intersection point C′ of two random locations using the following input parameters:

$$C' = f\ (h_a, k_a, r_a' + \Delta_a, h_b, k_b, r_b' + \Delta_b) \tag{17}$$

This error may be due to the influence of environmental noise, sound propagation, and other factors, resulting in inaccurate measurement distance and making the circle's radius larger or smaller. This error can even cause the circles to be disjointed, i.e., the two circles are entirely separated (as shown in Figure 11), or one circle is inside the other (as shown in Figure 12). We must account for and deal with these errors in practical applications to ensure our system models can effectively cope with uncertainties.

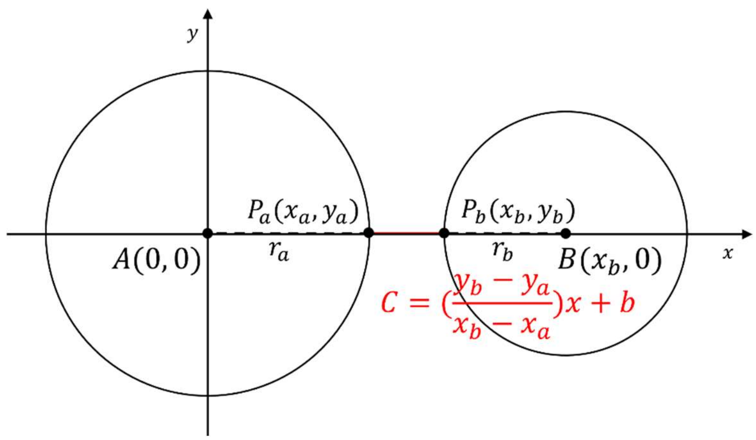

Figure 11: Special Case 1

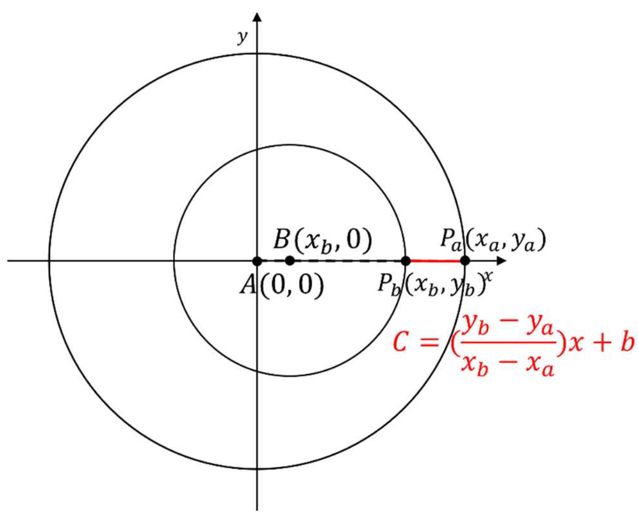

Figure 12: Special Case 2

4.4.4 Special Cases

In two particular cases, we cannot use the standard circle intersection method to determine the position of the ASR device because the two circles do not intersect. Suppose we determine that two circles do not produce any intersection. In that case, we will distinguish the two cases according to the rules of the special cases and use different methods to determine the best position.

Special case 1: As shown in Figure 11, when the distance between the centers of two circles is greater than the sum of their radii, that is, $d > r_a + r_b$. This means the two circles are far apart with a gap in between because the maximum distance between the centers of the two circles is the sum of the radii of the two circles. In this case, the boundaries of a circle define the farthest limit of all points within the circle, and if these boundaries do not overlap, there cannot be a common point between the two circles, i.e., there is no intersection.

Special case 2: As shown in Figure 12, when the distance between the centers of two circles is greater than the absolute value of the difference between their radii, that is, $d < abs(r_b - r_a)$. In this case, the smaller circle is completely inside the larger circle, so the radius of one circle must be larger than the radius of the other.

In this case, the most reasonable position is to find the point P on the two circles closest to the other circle and then find the line segment between these two points. All points on this line segment can be considered as the possible meeting locations between the two circle centers, which is the estimated location of the ASR device.

Among them (h, k) is the center point of the small circle, and r is the radius of the small circle. The result is the estimated position of the ASR device.

CHAPTER V

EXPERIMENTAL EVALUATION

This section describes the implementation details of how *SpyLoc* detects whether there is a hidden ASR device and evaluates it.

5.1 EXPERIMENT SETUP

Our evaluation considered three different ASR devices: Amazon Echo Dot, Google Nest Mini, and Apple HomePod Mini due to their popularity and user familiarity. The user first stimulates the ASR device by playing the wake-up word in the area of interest and then determines whether a hidden ASR device monitors the area by detecting changes in the number of WiFi traffic packets. MAC OUI is compared to find and monitor hidden ASR devices. Finally, the distance between the user-selected random location and the ASR device is calculated based on its wake-up rate, and the distance is used to find the exact location of the ASR device.

WiFi Packet Monitoring: We use Raspberry Pi 4 Model B and the airodump-ng tool running on Kali Linux to monitor and analyze the WiFi traffic statistics [64, 65]. We designed a Python script for network monitoring and data processing on Linux that executes shell commands using subprocess.Popen. The Raspberry Pi enters monitor mode and captures all transmitted packets within the WiFi band to aggregate traffic statistics for analysis. A smartphone can be used instead of a Raspberry Pi in *SpyLoc*, but a rooted phone is required [66].

Experiment Environments Setup: Two environments were tested (see Figure 13).

- **Bedroom:** We chose a bedroom about 130 inches * 117 inches for the experiment. The setup involved placing the device on a table and then it behind a book;
- **Conference Room:** We chose a conference room of about 133 inches * 285 inches for the experiment. The setup involved placing the device on a table and hiding it behind a book.

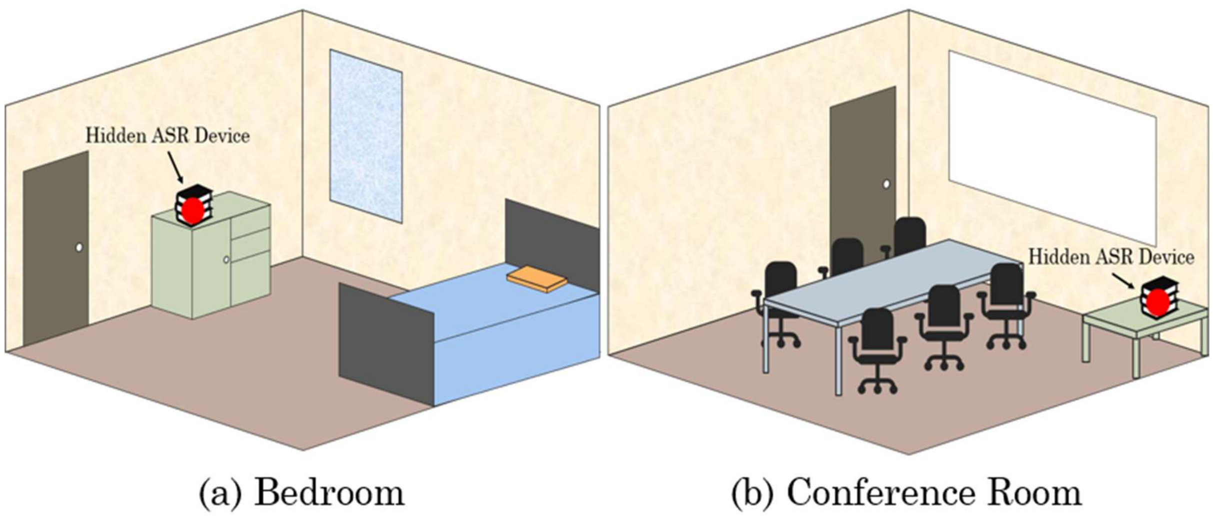

Figure 13: Experiment Environment Setup

Metrics: We use the following two metrics.

- **Localization Error:** This is measured based on the distance between the ASR device's estimated location and its corresponding true location.
- **Localization Time:** This is the time taken to obtain the accurate location of the ASR device.

S5.2 CASE STUDY

In this example, the hidden Amazon Echo Dot will be placed at the location marked in Figure 13. During the sample test, the user first needs to determine the presence of an ASR device monitoring the target area.

The user monitors the traffic within the test range of interest. In order to determine the existence of the hidden ASR device, *SpyLoc* requires the user to perform a stop-start-stop-start-stop (S5) action as follows: 1) The user remains quiet for a period of time to allow the device traffic to stabilize. 2) The user plays the wake-up words of the ASR devices listed in Table 1 at the current location using a volume of about 70 decibels. 3) The user stops again and waits for the device traffic to stabilize. 4) The user plays again. 5) User stops. S5 action causes unique patterns in WiFi traffic, as shown in Figure 4. We observed reduced traffic when the person remained quiet, but the traffic increased when the person started playing the wake-up word. When a trend is present, we determine the presence of hidden ASR devices.

We place the Echo Dot on the table and keep the phone parallel to the ASR device. We will randomly select two positions in the target area. The firthe st point is the origin, and the x position of the other point is the distance from the first point to the second point, and y is 0, where each inch represents one unit on the xy-axis. We will measure the wake-up probability at two points using different volume levels, and use the collected data to calculate the distance between the randomly selected location and the Echo Dot. Then, we can calculate the actual location of the Echo Dot.

In our local testing, *SpyLoc* successfully identified traffic from ASR devices in all 10 trials. By comparing the error between the ASR device's estimated x and y coordinate and the actual value and calculating the localization error, we can see in Figure 14 that the localization error is always less than 0.53 inches. In most cases, the localization error will be closer to the error of the x or y with the larger error. The average localization error for x, y, and the ASR device is 0.44 inches, 0.37 inches, and 0.53 inches, respectively. Taking the size of the Amazon Echo Dot (3.9 inches * 3.9 inches * 3.5 inches) as a reference shows the high accuracy of *SpyLoc* in the local environment. Figure 15 shows the localization time for each test. Because the same decibel does not always awaken the ASR device, we need to wake it up 10 times, resulting in a slightly longer localization time, about 303 seconds.

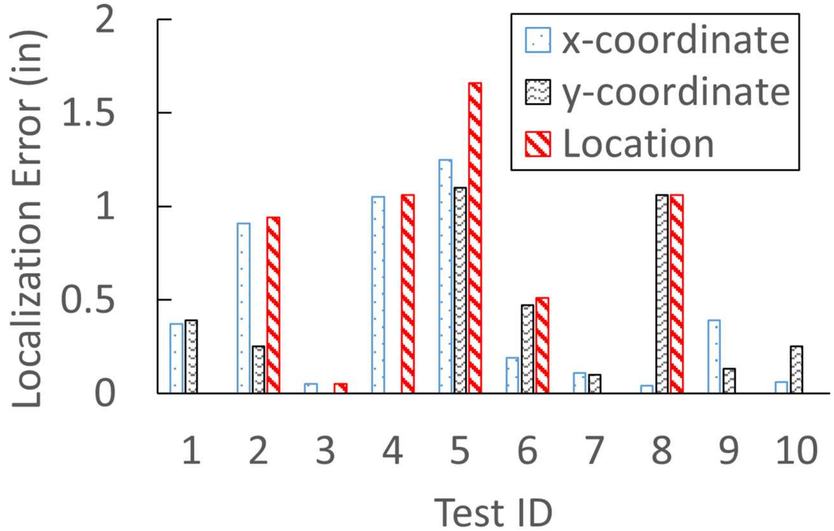

Figure 14: Localization error analysis in all 10 trials.

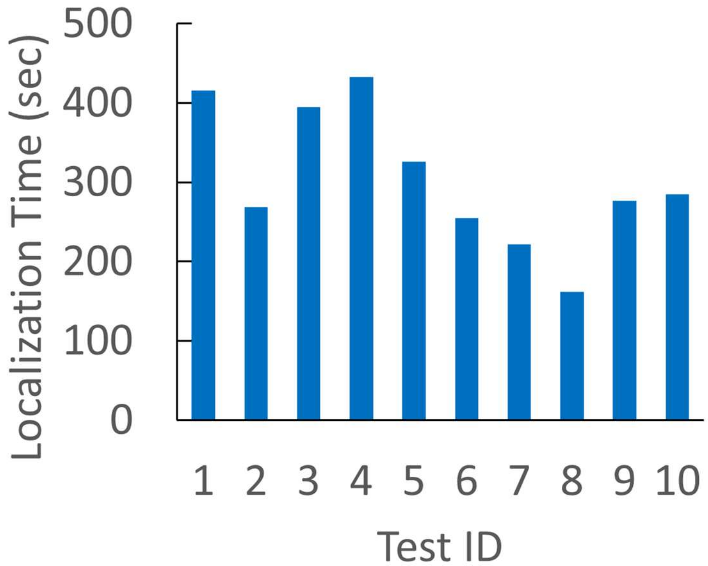

Figure 15: Localization time analysis in all 10 trials.

Impact of Noise: In Environment A, where the room is kept quiet, the ambient volume is 28.85 dB. However, Environment B has a higher ambient volume of 49.25 dB, indicating increased noise. Similarly, we perform 30 attempts with *SpyLoc* to locate hidden ASR devices.

We observe in Figure 16 that the environment's noise level may impact the positioning accuracy. In Environment A, the average error is 4.48 inches, whereas in Environment B, it increases to 6.25 inches. This low-noise environment (Environment A) can enhance the recognition accuracy of ASR devices, as lower background noise facilitates the device's capture and helps understand the user's voice commands. In contrast, in a high-noise environment (Environment B), the noise hampers the clarity and quality of the voice signal, making it more challenging for the device to recognize and understand the user's voice commands accurately.

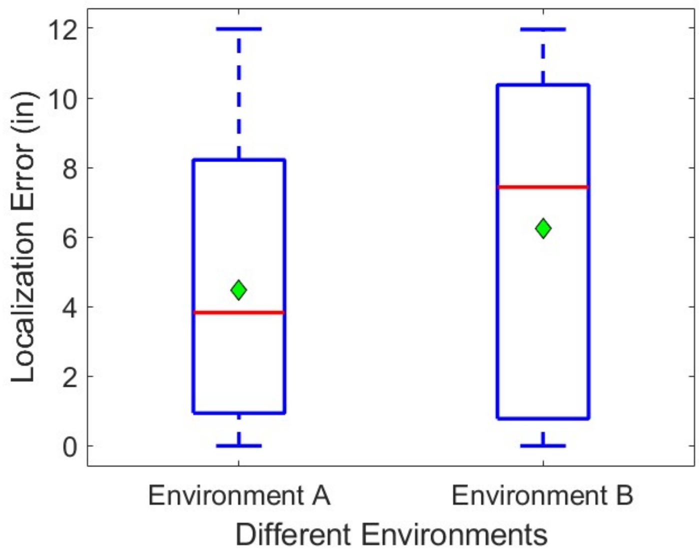

Figure 16: Different environments localization error of Amazon Echo Dot.

5.3 OVERALL PERFORMANCE

We evaluated the performance of all ASR devices in a specific experimental room environment (see Figure 13). To locate ASR devices across various locations and environments, we conducted 30 trials for each ASR device in two environments, totaling 180 trials (30 trials * 3 devices * 2 environments). We calculated the localization error in each trial and recorded the localization time.

We calculated and analyzed three ASR devices' average localization error and localization time in two different noise environments. The average localization error of ASR devices located by our method in Environment A is less than 4.93 inches, and the average localization time can be controlled within 303.15 seconds, which shows the accuracy and feasibility of our method. In

Environment B, our average localization error is less than 5.64 inches, which is only 0.71 inches different from the error in Environment A, indicating that the system has high robustness in other environments. This demonstrates that environmental noise has minimal impact on the experimental results, and the experiment can maintain consistent output under different noises, highlighting its anti-interference ability and adaptability.

Figures 17 and 18 detail the performance of various ASR devices in the experimental environment. In Environment A, the average localization error of Amazon Echo Dot is 4.48 inches, the average localization time is 310 seconds; the average localization error of Google Nest Mini is 5.8 inches, the average localization time is 357 seconds; and the average localization error of Apple HomePod Mini is 4.59 inches, the average localization time is 243 seconds. In Environment B, the average localization errors were 6.25 inches, 5.06 inches, and 5.6 inches for the Amazon Echo Dot, Google Nest Mini, and Apple HomePod Mini, respectively, with localization times of 289 seconds, 346 seconds, and 286 seconds. The minimum localization error can reach 0. Notably, in the same environment, Google's voice assistant device showed a relatively large localization error and a long localization time, which may be related to the fact that the wake-up word of Google's voice assistant device consists of two words, while the other two ASR devices are one word (see Table 1). These results highlight that our system achieves high localization accuracy and quick response times, even under challenging conditions.

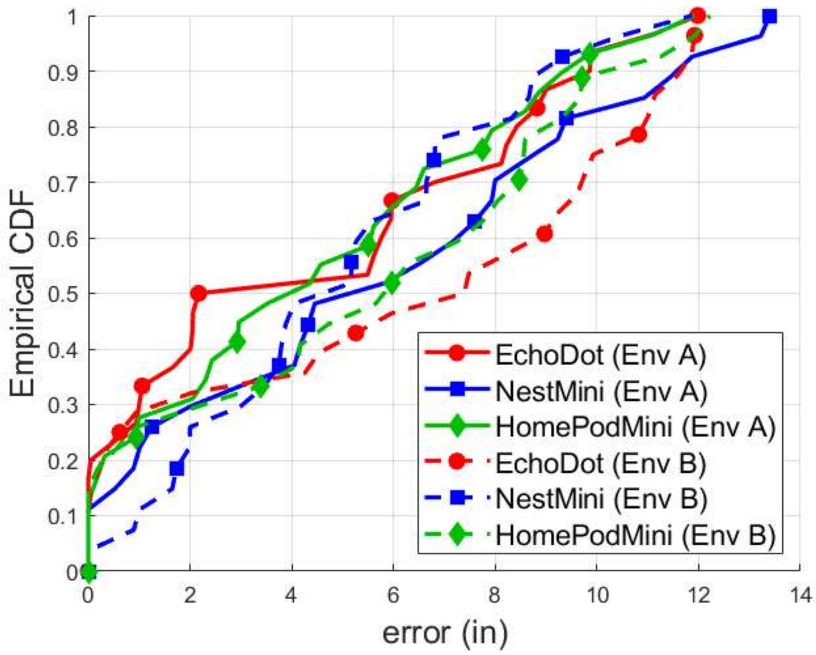

Figure 17: Overall devices localization error in environments A and B.

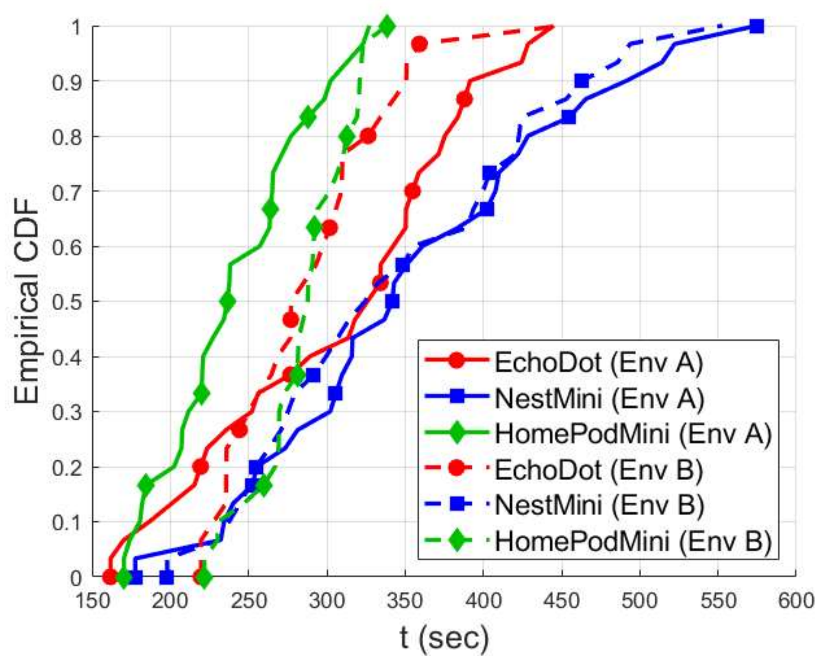

Figure 18: Overall devices localization time in environments A and B.

5.4 USER STUDY

We recruited 10 volunteers (5 self-identified females and males) and asked each to use *SpyLoc* to find hidden ASR devices randomly deployed in a conference room. Each participant randomly selected a device to locate (Amazon Echo Dot, Google Nest Mini, and Apple Home Pod Mini) for 10 attempts in the current environment. We compared the localization results with the proper position of the ASR device to quantify the localization error.

Figure 19 shows the localization errors obtained by different users in the same environment. We can observe that the average error range of different users of all users ranges from 0.34 to 5.15 inches, the average error of all data is around 4.28 inches, and the minimum localization error is 0. User localization results closely align with experimental test results, demonstrating consistent localization accuracy across trials.

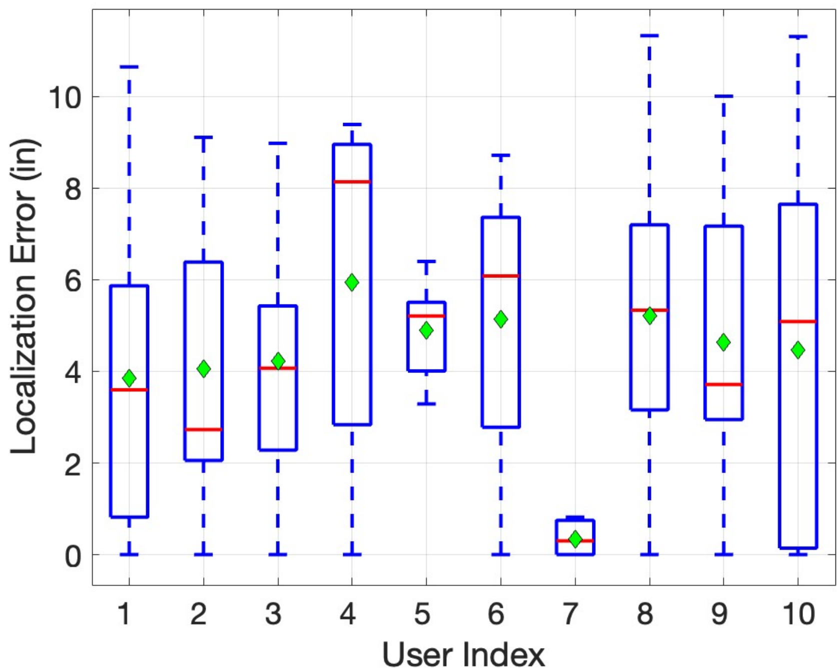

Figure 19: Localization error for all 10 users.

The average localization time varies depending on the device selected by the user. Table 2 summarizes each device's average, minimum, and maximum localization times. We found that the average localization time for all users, except for a few outliers, was less than 300 seconds for

Amazon and Apple's voice assistant devices and less than 412 seconds for Google's voice assistant devices. This verifies the practicality of the proposed ASR device positioning strategy.

Table 2: Localization time for different devices by users.

Brand Name	Localization Time (seconds)		
	Average	Minimum	Maximum
Amazon	228	145	317
Google	309	194	457
Apple	238	164	329

CHAPTER VI

DISCUSSION

6.1 LIMITATIONS

Localization Time. We play the wake-up word multiple times at varying volumes, aiming to identify the volume level that can trigger the hidden ASR device. After each activation, we wait several seconds before attempting to wake up the ASR device again. This approach allows us to collect data to analyze the system's wake-up probability and infer the location of the ASR device. Such a testing process involves repeatedly playing the wake-up words and meticulously recording and analyzing the data requiring considerable time.

Sound Reflection. When sound propagates, it encounters obstacles such as walls and furniture, leading to reflections. These reflections may cause the sound wave to lose energy, especially after multiple reflections, resulting in sound signal attenuation. This degree of attenuation increases with the transmission distance, weaking the sound signal by the time it reaches the ASR device. As a result, the ASR device may struggle to accurately recognize the diminished signal potentially impacting its ability to respond correctly.

Sound reflection also causes sound waves to spread widely in space, mixing the direct sound and reflected sound. This combination of sound waves can reduce the clarity of the signal, making it more challenging for devices to recognize and accurately interpret the sound [67]. In closed or semi-enclosed spaces, multiple sound reflections can also produce echo and reverberation effects, further interfering with the performance of ASR devices. The effectiveness of an ASR system relies heavily on the quality of the received sound signal. Signal attenuation or distortion caused by these effects can significantly hinder the system's ability to accurately recognize speech, leading to an increase in recognition errors.

Sound-absorbing panels can be installed to reduce reflections. However, this requires users to purchase extra sound-absorbing materials, thereby increasing the cost of ASR device localization. Moreover, even with sound-absorbing materials installed on walls and ceilings, other reflective surfaces in the room, such as glass windows, large metal objects, or hardwood furniture, can still generate sound reflections. These reflective surfaces can compromise the effectiveness of acoustic treatments, meaning that reflection issues may persist despite efforts to improve the room's acoustic environment.

6.2 FUTURE WORK

While the results of the proposed ASR localization technique are promising, several research directions warrant further exploration.

One notable avenue is enhancing the scalability of the proposed localization technique by integrating it with other localization methods, such as WiFi localization, which aims to discover a wireless device by leveraging the characteristics of the wireless signal the device generates. Combining these approaches could expand the range of ASR device localization. For instance, WiFi localization could be employed to determine an approximate location, followed by the application of the proposed technique to refine the estimate and minimize localization errors. This hybrid approach would offer a more comprehensive and accurate method for identifying ASR device positions across larger areas.

Second, the results of this study are tested on only three ASR devices from the most common brands in the market: Amazon, Google, and Apple. Future work should expand testing to include a wider range of devices with voice assistant functionality, such as smartphones, laptops, and other smart devices. This expansion would offer a more comprehensive evaluation of the proposed technique's effectiveness across diverse device types and brands, ensuring its broader applicability and robustness in real-world scenarios.

Third, although this study primarily focused on localizing ASR devices that rely on wake-up words for activation, future research could expand this framework to encompass devices that utilize alternative remote activation mechanisms, such as gesture recognition and proximity sensors. Exploring such devices would help generalize the findings, increase the versatility of the proposed technique, and extend its impact to a broader range of smart technologies with diverse activation methods.

CHAPTER VI

CONCLUSION

ASR devices have become ubiquitous in recent years. and they are now integrated into many aspects of daily life, including virtual assistants. However, these devices may transmit recorded audio data to the cloud for processing without the user's awareness, which may be intercepted or abused without authorization. In this thesis, we propose *SpyLoc*, a new defense technology designed to prevent unauthorized audio recording by detecting hidden ASR devices in the environment. The core idea of *SpyLoc* is to locate the device by stimulating the ASR device and monitoring the changes in wireless traffic generated during the response. By analyzing these traffic changes, *SpyLoc* can calculate the device's wake-up probability and pinpoint the device's physical location accordingly. This method leverages the fact that ASR devices must communicate with a cloud server when activated. By inducing and monitoring this communication, the location of the ASR device can be inferred without direct contact. Our real-world evaluation results demonstrate the capability of *SpyLoc* to accurately position three mainstream ASR devices in two distinct scenarios. *SpyLoc* represents a promising approach to mitigating real threats to privacy in smart home environments.

This thesis comes from a conference paper in submission.

REFERENCES

[1] Federica Laricchia. 2022. Number of digital voice assistants in use worldwide from 2019 to 2024 (in billions). https://www.statista.com/statistics/973815/worldwidedigital-voice-assistant-in-use/.

[2] Bergur Thormundsson. 2023. Number of voice assistant users in the United States from 2022 to 2026 (in millions). https://www.statista.com/statistics/1299985/voice-assistant-users-us/.

[3] Chen Yan, Xiaoyu Ji, Kai Wang, Qinhong Jiang, Zizhi Jin, and Wenyuan Xu. A survey on voice assistant security: Attacks and countermeasures. *ACM Computing Surveys*, 55(4), Nov 2022.

[4] Tom Bolton, Tooska Dargahi, Sana Belguith, Mabrook S. Al-Rakhami, and Ali Hassan Sodhro. On the security and privacy challenges of virtual assistants. *Sensors*, 21(7), 2021.

[5] Amazon. 2023. Amazon Introduces Four All-New Echo Devices, Sales of Alexa-Enabled Devices Surpass Half a Billion. https://press.aboutamazon.com/2023/5/amazon-introduces-four-all-new-echo-devices-sales-of-alexa-enabled-devices-surpass-half-a-billion.

[6] Federica Laricchia. 2022. Total smart home devices supported by various voice assistants in 2018 and 2019. https://www,statista.com/statistics/933551/worldwidevoice-assistant-supported-smart-home-devices/.

[7] Zhaohe (John) Zhang, Edwin Yang, and Song Fang. Commandergabble: A universal attack against asr systems leveraging fast speech. In *Proceedings of the 37th Annual Computer Security Applications Conference (ACSAC '21)*, page 720–731, New York, NY, USA, 2021. Association for Computing Machinery.

[8] Deepak Kumar, Riccardo Paccagnella, Paul Murley, Eric Hennenfent, Joshua Mason, Adam Bates, and Michael Bailey. Emerging threats in internet of things voice services. *IEEE Security Privacy*, 17(4):18–24, 2019.

[9] Richard Mitev, Anna Pazii, Markus Miettinen, William Enck, and Ahmad-Reza Sadeghi. Leakypick: IoT audio spy detector. In *Proceedings of the 36th Annual Computer Security Applications Conference (ACSAC '20)*, page 694–705, New York, NY, USA, 2020. Association for Computing Machinery.

[10] Lingkun Li, Manni Liu, Yuguang Yao, Fan Dang, Zhichao Cao, and Yunhao Liu. Patronus: preventing unauthorized speech recordings with support for selective unscrambling. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems (SenSys '20)*, page 245–257, New York, NY, USA, 2020. Association for Computing Machinery.

[11] Richard Mitev, Markus Miettinen, and Ahmad-Reza Sadeghi. Alexa lied to me: Skill-based man-in-the-middle attacks on virtual assistants. In *Proceedings of the 2019 ACM Asia Conference*

on Computer and Communications Security (Asia CCS '19), page 465–478, New York, NY, USA, 2019. Association for Computing Machinery.

[12] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*, page 103–117, New York, NY, USA, 2017. Association for Computing Machinery.

[13] Rose Ceccio, Sophie Stephenson, Varun Chadha, Danny Yuxing Huang, and Rahul Chatterjee. Sneaky spy devices and defective detectors: the ecosystem of intimate partner surveillance with covert devices. In *Proceedings of the 32nd USENIX Conference on Security Symposium (SEC '23)*, USA, 2023. USENIX Association.

[14] Matt Day, Giles Turner, and Natalia Drozdiak. Amazon Workers Are Listening to What You Tell Alexa. https://www.bloomberg.com/news/articles/2019.04-10/is-anyone-listening-to-you-on-alexa-a-global-team-reviews-audio.

[15] Aimee Picchi. 2019. Amazon workers are listening to what you tell Alexa. https://www.cbsnews.com/news/amazon-workers-are-listening-towhat-you-tell-alexa/.

[16] Jordan Valinsky. 2019. Amazon reportedly employs thousands of people to listen to your Alexa conversations. https://www,cnn.com/2019/04/11/tech/amazon-alexa-listening/index.html.

[17] Apple. 2023. Improve Siri and Dictation Privacy. https://www.apple.com/legalprivacy/data/en/improve-siri-dictation/.

[18] Tim Verheyden, Denny Baert, Lente Van Hee, and Ruben Van Den Heuvel. 2019. Google employees are eavesdropping, even in your living room, VRT NWS has discovered. https://www.vrt.be/vrtnws/en/2019/07/10/google-employees-are-eavesdropping-even-in-flemish-living-rooms/.

[19] Antoni Rogalski. Infrared detectors: status and trends. *Progress in quantum electronics*, 27(2-3):59–210, 2003.

[20] Ruochen Zhou, Xiaoyu Ji, Han Chen, Chen Yan, and Wenyuan Xu. Detecting hidden voice recorders via ADC electromagnetic radiation. *ACM Transactions on Sensor Networks*, October 2024.

[21] Xuxin Lin, Jianwen Gan, Chaohao Jiang, Shuai Xue, and Yanyan Liang. Wi-Fi-based indoor localization and navigation: A robot-aided hybrid deep learning approach. *Sensors*, 23(14):6320, 2023.

[22] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using WiFi. In *Proceedings of the 2015 ACM conference on special interest group on data communication* (pp. 269-282), August 2015.

[23] Zahraa A Hamza, Mahmood F Mosleh, and Faeza A Abed. Wi-Fi positioning utilizing triangulation strategies and AOA in indoor environments. In *Proceedings of the 2nd International Multi-Disciplinary Conference Theme: Integrated Sciences and Technologies*, page 64, 2022.

[24] Siri Team. 2017. Hey Siri: An On-device DNN-powered Voice Trigger for Apple's Personal Assistant. https://machinelearning.apple.com/research/hey-siri.

[25] Yan He, Qiuye He, Song Fang, and Yao Liu. 2021. MotionCompass: pinpointing wireless camera via motion-activated traffic. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21)*. Association for Computing Machinery, New York, NY, USA, 215–227.

[26] Apple. 2024. Sensor & Usage Data & Privacy. https://www.apple.com/legal/privacy/data/en/sensor-usage-data/.

[27] Rameez Raja Kureshi and Bhupesh Kumar Mishra. A comparative study of data encryption techniques for data security in the IoT device. In *Internet of Things and Its Applications: Select Proceedings of ICIA 2020*, pages 451–460, Singapore, 2022. Springer Nature Singapore.

[28] 2024. Alexa Voice Interoperability Human Interface Design Guide. Amazon Developer. https://developer.amazon.com/en-US/alexa/voice-interoperability/design-guide/human-interface-design

[29] Amazon.2024. Alexa, Echo Devices, and Your Privacy. https://www.amazoncom/gp/help/customer/display.html?nodeId=GVP69FUJ48X9DK8V.

[30] Mia Chiquier, Chengzhi Mao, and Carl Vondrick. Real-time neural voice camouflage. *arXiv preprint arXiv:2112.07076*, 2021.

[31] Holly Evarts. Concerned your smartphone is spying on you?. https://www.engineering.columbia.edu/news/block-smartphone-microphone-speech-recognition-spying.

[32] Qiuye He, Edwin Yang and Song Fang. 2024. A Survey on Human Profile Information Inference via Wireless Signals. In *IEEE Communications Surveys & Tutorials*, vol. 26, no. 4, 2577-2610.

[33] Peng Huang, Yao Wei, Peng Cheng, Zhongjie Ba, Li Lu, Feng Lin, Fan Zhang, and Kui Ren. Infomasker: Preventing eavesdropping using phoneme-based noise. In *NDSS*, 2023.

[34] Antônio J. Pinheiro, Jeandro de M. Bezerra, Caio A.P. Burgardt, and Divanilson R. Campelo. Identifying IoT devices and events based on packet length from encrypted traffic. *Computer Communications*, 144:8–17, 2019.

[35] Yan He, Qiuye He, Song Fang, and Yao Liu. 2023. When Free Tier Becomes Free to Enter: A Non-Intrusive Way to Identify Security Cameras with no Cloud Subscription. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23)*. Association for Computing Machinery, New York, NY, USA, 651–665.

[36] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. Peek-a-boo: i see your smart home activities, even encrypted! In P*roceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '20)*. ACM, July 2020.

[37] David Caputo, Luca Verderame, Alessio Merlo, Andrea Ranieri, and Luca Caviglione. Are you (google) home? detecting users' presence through traffic analysis of smart speakers. In *Italian Conference on Cybersecurity*, 2020.

[38] Chenggang Wang, Sean Kennedy, Haipeng Li, King Hudson, Gowtham Atluri, Xuetao Wei, Wenhai Sun, and Boyang Wang. Fingerprinting encrypted voice traffic on smart speakers with deep learning. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 254-265.

[39] Yan He, Qiu He, Song Fang and Yao Liu. 2024. Precise Wireless Camera Localization Leveraging Traffic-Aided Spatial Analysis. In *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, 7256-7269.

[40] Jianghan Mao, Chenyu Wang, Yanhui Guo, Guoai Xu, Shoufeng Cao, Xuanwen Zhang, and Zixiang Bi. A novel model for voice command fingerprinting using deep learning. *Journal of Information Security and Applications*, 65:103085, 2022.

[41] Lea Schonherr, Maximilian Golla, Thorsten Eisenhofer, Jan Wiele, Dorothea Kolossa, and Thorsten Holz. Exploring accidental triggers of smart speakers. *Computer Speech & Language*, 73:101328, 2022.

[42] Andreas Zehetner, Martin Hagmuller, and Franz Pernkopf. Wake-up-word spotting for mobile systems. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 1472–1476, 2014.

[43] Yixin Gao, Yuriy Mishchenko, Anish Shah, Spyros Matsoukas, and Shiv Vitaladevuni. Towards data-efficient modeling for wake word spotting. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7479–7483, 2020.

[44] Tiwari, Vibha. MFCC and its applications in speaker recognition. *International journal on emerging technologies 1*, no. 1 (2010): 19-22.

[45] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.

[46] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[47] Qiuye He and Song Fang. 2023. Phantom-CSI Attacks against Wireless Liveness Detection. In Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '23). Association for Computing Machinery, New York, NY, USA, 440–454.

[48] 2024. Turn off Classic Voice Control on your iPhone, iPad, or iPod touch. https://support.apple.com/en-us/119836.

[49] Chen Wang, S Abhishek Anand, Jian Liu, Payton Walker, Yingying Chen, and Nitesh Saxena. Defeating hidden audio channel attacks on voice assistants via audio-induced surface vibrations. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 42–56, 2019.

[50] Yukang Yan, Chun Yu, Yingtian Shi, and Minxing Xie. Privatetalk: Activating voice input with hand-on-mouth gesture detected by Bluetooth earphones. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 1013–1020, 2019.

[51] Vaishnavi Mande, Abraham Glasser, Becca Dingman, and Matt Huenerfauth. Deaf users' preferences among wake-up approaches during sign-language interaction with personal assistant devices. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2021.

[52] Veton Kepuska. Wake-up-word speech recognition. *Speech Technologies*, pages 237–262, 2011.

[53] Reza Yazdani, Jose-Maria Arnau, and Antonio Gonzalez. A low-power, high-performance speech recognition accelerator. *IEEE Transactions on Computers*, 68(12):1817–1831, 2019.

[54] IEEE standards association - organizationally unique identifier (OUI) public listing. https://standards-oui.ieee.org/oui/oui.txt.

[55] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S. Cardoso, and Frank Piessens. Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16)*, page 413–424, New York, NY, USA, 2016. Association for Computing Machinery.

[56] Liang Xiao, Xiaoyue Wan, Xiaozhen Lu, Yanyong Zhang, and Di Wu. IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?. *IEEE Signal Processing Magazine*, 35(5):41–49, 2018.

[57] Jeremy Martin, Travis Mayberry, Collin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Erik C. Rye, and Dane Brown. A study of MAC address randomization in mobile devices and when it fails. *arXiv preprint arXiv:1703.02874*.

[58] Jeremy Martin, Erik Rye, and Robert Beverly. Decomposition of MAC address structure for granular device inference. In *Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC '16)*, page 78–88, New York, NY, USA, 2016. Association for Computing Machinery.

[59] Santiago Fernandez, Alex Graves, and Jurgen Schmidhuber. An application of recurrent neural networks to discriminative keyword spotting. In *International conference on artificial neural networks*, pages 220–229. Springer, 2007.

[60] T. Rossing, F. Dunn, W.M. Hartmann, D.M. Campbell, and N.H. Fletcher. Springer Handbook of Acoustics. *Springer Handbooks*. Springer New York, 2015.

[61] Benjamin K Dichter, Jonathan D Breshears, Matthew K Leonard, and Edward F Chang. The control of vocal pitch in human laryngeal motor cortex. *Cell*, 174(1):21–31, 2018.

[62] Lasse Jakobsen, Jakob Christensen-Dalsgaard, Peter Møller Juhl, and Coen PH Elemans. How loud can you go? physical and physiological constraints to producing high sound pressures in animal vocalizations. *Frontiers in Ecology and Evolution*, 9:657254, 2021.

[63] Ping Tang, Xiangming Li, and Shuai Wang. A differential weighted accumulation algorithm using variable sliding window. *IEEE Access*, 6:21220–21230, 2018.

[64] Raspberry Pi Foundation. Raspberry Pi 4 model B. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/.

[65] Aircrack-ng Team. Airodump-ng - wireless network sniffer and detector. https://www.aircrack-ng.org/doku.php?id=airodump-ng.

[66] San-Tsai Sun, Andrea Cuadros, and Konstantin Beznosov. Android rooting: Methods, detection, and evasion. In *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, pages 3–14, 2015.

[67] Lauri Savioja and U. Peter Svensson. Overview of geometrical room acoustic modeling techniques. *The Journal of the Acoustical Society of America*, 138(2):708–730, 08 2015.