MULTI-VECTOR TRACKING OF WIFI AND ZIGBEE DEVICES

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science

by
Calvin Laverty
June 2019
© 2019
Calvin Laverty
ALL RIGHTS RESERVED
COMMITTEE MEMBERSHIP

TITLE: Multi-Vector Tracking of WiFi and Zigbee Devices

AUTHOR: Calvin Laverty

DATE SUBMITTED: June 2019

COMMITTEE CHAIR: Bruce DeBruhl, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Hugh Smith, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: John Bellardo, Ph.D.
Professor of Computer Science
ABSTRACT

Multi-Vector Tracking of WiFi and Zigbee Devices

Calvin Laverty

Location privacy preservation has shifted to the forefront of discussions about next generation wireless networks. While pseudonym-changing schemes have been proposed to preserve an individual’s privacy, simulation has shown that new association attack models render these schemes useless. The major contribution of this thesis is the implementation of a tracking network with commodity hardware on the California Polytechnic State University campus which leverages the combination of deanonymization strategies on captured wireless network data to show the effectiveness of a pseudonym-changing scheme for wireless identification across WiFi and Zigbee protocols.
ACKNOWLEDGMENTS

Thanks to:

- My parents for supporting me through all of my academic endeavors
- Shannon, Allison, and Kirsten for being the best sisters a brother could ask for
- Dr. DeBruhl for guiding me through my thesis project and fostering a love for research
- Aaron Otis, Collin Kenner, Ethan Goldfarb, and Justin Nguyen for all of their help on this project and being tracking guinea pigs
- All of my professors that spent countless hours debugging with me
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CHAPTER</strong></td>
<td></td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Social Implications</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Privacy Protection with Pseudonyms</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Basic Tracking Implementation Overview</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Contributions</td>
<td>5</td>
</tr>
<tr>
<td>2 Background</td>
<td>6</td>
</tr>
<tr>
<td>2.1 802.11 - WiFi</td>
<td>6</td>
</tr>
<tr>
<td>2.2 802.15.4 - Zigbee</td>
<td>8</td>
</tr>
<tr>
<td>2.3 LTE</td>
<td>9</td>
</tr>
<tr>
<td>3 Related Works</td>
<td>11</td>
</tr>
<tr>
<td>3.1 Jamming Attacks</td>
<td>11</td>
</tr>
<tr>
<td>3.2 Paging Attacks</td>
<td>11</td>
</tr>
<tr>
<td>3.3 Location Privacy</td>
<td>13</td>
</tr>
<tr>
<td>3.4 Pseudonym Association Attack Model</td>
<td>14</td>
</tr>
<tr>
<td>4 LTE Tracking</td>
<td>16</td>
</tr>
<tr>
<td>4.1 Theoretical Implementation</td>
<td>17</td>
</tr>
<tr>
<td>4.2 SRS-LTE</td>
<td>17</td>
</tr>
<tr>
<td>4.3 Paging Attack Implementation - A Building Block</td>
<td>18</td>
</tr>
<tr>
<td>4.4 Complete UE Tracking</td>
<td>19</td>
</tr>
<tr>
<td>4.5 Experimentation Setup</td>
<td>19</td>
</tr>
</tbody>
</table>
4.6 Results ................................................................. 20
  4.6.1 Using the SRS-LTE Platform for Tracking ............... 20
  4.6.2 Paging Attacks: Successes and Failures .................. 21
  4.6.3 Lack of LTE Network Control and Information ............ 21
4.7 Conclusion ......................................................... 22
5  De-Anonymization Algorithm Implementation .................. 23
  5.1 Data Matrices .................................................... 24
  5.2 Association and Disassociation Classifications ............. 26
  5.3 Exit-to-Enter Attack .......................................... 28
  5.4 Transitive Property ........................................... 31
  5.5 Boosting Processes ............................................ 32
  5.6 Confidence Threshold ......................................... 34
  5.7 Path Reconstruction ........................................... 34
6  Multi-Vector Listener Implementation ........................... 35
  6.1 802.11 Tracking ............................................... 35
  6.2 802.15.4 Tracking ............................................. 36
  6.3 Controlled Tracking Experimentation Setup ................ 37
7  Multi-Vector Target Device Implementation ..................... 38
  7.1 802.11 Tracking ............................................... 39
  7.2 802.15.4 Tracking Implementation ............................ 40
8  Experiment Environment .......................................... 41
  8.1 Environment Overview ........................................ 41
  8.2 Listener Setup and Placement ................................ 42
    8.2.1 Test Placement ........................................... 44
  8.3 Data Collection Listener Coverage ............................ 45
<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Basic EnodeB, UE, and MME Connections</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>Attack Model Overview</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>Zigbee to Zigbee Matrix</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>Initial WiFi to Zigbee Matrix</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>Example Known Association</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>Example Possible Association</td>
<td>27</td>
</tr>
<tr>
<td>7</td>
<td>Example Known Disassociation</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>Observation Intervals for the Exit-to-Enter Attack</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>Exit-to-Enter Bipartite Matching</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>Boosting Process</td>
<td>33</td>
</tr>
<tr>
<td>11</td>
<td>Target Device Implementation</td>
<td>38</td>
</tr>
<tr>
<td>12</td>
<td>Leveraging the Hidden Terminal Problem</td>
<td>39</td>
</tr>
<tr>
<td>13</td>
<td>Listener Container and Protection</td>
<td>42</td>
</tr>
<tr>
<td>14</td>
<td>Listener Implementation</td>
<td>43</td>
</tr>
<tr>
<td>15</td>
<td>Expected Listener WiFi Coverage for Testing</td>
<td>44</td>
</tr>
<tr>
<td>16</td>
<td>Expected Coverage of Zigbee Measurements from GPS Coordinates</td>
<td>46</td>
</tr>
<tr>
<td>17</td>
<td>Expected Listener WiFi Coverage for Data Collection</td>
<td>47</td>
</tr>
<tr>
<td>18</td>
<td>Measured Listener WiFi Coverage for Data Collection</td>
<td>47</td>
</tr>
<tr>
<td>19</td>
<td>Expected Listener Zigbee Coverage for Data Collection</td>
<td>48</td>
</tr>
<tr>
<td>20</td>
<td>Measured Listener Zigbee Coverage for Data Collection</td>
<td>48</td>
</tr>
<tr>
<td>21</td>
<td>Movement Matrix Adaptation</td>
<td>55</td>
</tr>
<tr>
<td>22</td>
<td>Total Pseudonym Counts for Corresponding Switching Rate</td>
<td>57</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>W2D Association Success vs. Pseudonym-Switching Rate</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>Confidence Threshold Variation</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>Precision and Recall Results for Varying Delta</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>Exit-to-Enter Training Phase Variation Precision</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>Exit-to-Enter Training Phase Variation Recall</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>Reconstructed Path vs. Actual Path</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>Tracking Information Gained from a Single Vector</td>
<td>68</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

The mobile age has taken over the globe. It is common for an individual to have a mobile device in their pocket and wireless networks are widely available to these consumers. As of 2017, 68.4% of the U.S. population owns a smartphone [9]. Each of these mobile devices utilize common wireless transmissions such as WiFi, Bluetooth, Zigbee, GSM, and LTE.

In this paper, we utilize simulated tracking techniques that rely on the communication protocols that permeate wireless networks in everyday life. While this experiment is controlled to uniform targeted devices, the hope is to bring the topic of location privacy to the forefront of discussions when considering the implications of new technologies and what privacy an end-user is giving up by simply carrying a mobile device.

1.1 Social Implications

In the growing technology age, we see these mobile device replacing other stationary technologies. Workers can respond to emails on their bus ride home. Longtime friends that live in different places can start a video chat with just the push of a button [14]. Groceries stores are met with the new rival of Amazons Prime Pantry [13]. More time is spent in the virtual world than ever before and continues to grow in way that were not imagined in the past decade. As of 2016, nearly 80% of all social media browsing is done on mobile devices [41]. In 2017, those who owned mobile devices typically spend around 5 hours a day on their mobile devices [36].
The concern of privacy is at the forefront of the technology age. We see various privacy scandals like that of Facebook’s Cambridge Analytica [27]. Users have been leery of using some of Google’s services, even if they have decided to opt out of being tracked [8]. Lastly, blogs of recent privacy breaches such as Coffee Meets Bagel and Instagram are kept busy with the non-stop announcements that companies disclose to their users [43]. With privacy breaches a common occurrence, users hope to see a shift to a greater emphasis on the protection of their data.

Unbeknownst to most users, devices that are put in a sleep mode continue to utilize wireless communication protocols. This is the case with Wifi as discussed in [19]. The issue of location privacy has been growing alongside the increasing use of mobile devices and what information they leak about the user [23, 28, 38]. Location privacy is a concern that if the user is using a piece of technology, can the user be tracked as they move from place to place based on the unique signals that are given off by the device that they are carrying. This has been shown to be the case as tech giant Google has been tracking users even when location services on Android phones are off [35].

One such example to consider is a worker who takes the same route to work every day and passes the same landmarks. An adversary can complete a device “profile” through a combination of determining common times a device passed significant locations or unique signatures of the tracked device. Once this is complete, the attacker can set up probing stations to monitor a worker to determine their progress on their daily commute. As a consequence, this allows an adversary to take advantage of the worker not being home to perform malicious actions.

Previous work has been done in location tracking by using a single-vector signal source as a location indicator. This previous work discusses that a network of devices composed of some percentage of the population in a condensed area can track a user
through their own probing connections [30]. Since most mobile devices are equipped with WiFi, this is a prime target for adversaries to use for location tracking. Other work has focused on the idea of analog fingerprinting of devices. This idea is centered on the fact that different manufacturers have variations in implementation in either software or hardware. With enough data on behavior, an attacker can analyze the outgoing signal and timing and narrow the field of possible devices the target could be using. From here, the attacker can try to exploit and extrapolate this knowledge in order to track users targets [17]. This would allow for adversaries to profile certain devices and cross-reference the targets owned device to the profile obtained from the fingerprinting process.

1.2 Privacy Protection with Pseudonyms

Previous work has been done to use changing pseudonyms and mix zones (locations where devices can swap pseudonyms) in regards to a single tracking vector to maintain anonymity when using wireless communications [16, 37, 24]. These papers explore the application of pseudonym-based communication and their ability to thwart various tracking attacks. While there is more work to be done in this field, this thesis employs past techniques used, in addition to association across multiple tracking vectors in order to effectively de-anonymize target devices.

1.3 Basic Tracking Implementation Overview

This thesis is a proof of concept that, while already owning a set of unique identifiers across multiple communication protocols for a set of devices, one can associate identities across protocols in a common space using the periodic signals emitted by the various protocols on common mobile devices used today. This project utilizes
commodity chipsets for WiFi in conjunction with Zigbee. A ‘profile’ is built from the devices, in which a proof of concept implementation detects, collects, then analyzes these forms of communication to identify target devices and correlate across the different communication vectors.

A simplified tracking implementation is explored using a combination of the 802.11b Wifi protocol in conjunction with the 802.15.4 Zigbee protocol in order to associate addresses from one protocol to another across multiple listener probes. This uses the same concept of target devices periodically emitting beacons in either protocol that the listeners can sniff and then analyze. This is discussed more in the related works section. This proof of concept implementation detects, collects, and then analyzes the communications across the multiple vectors.

Over the period of four weeks, data is collected from the various probes and analyzed in two separate data sets. The first will be referred to as the ”initial data set” and the second will be referred to as the ”testing data set”.

The initial data set gives insight to the general behavior of the target devices and the effectiveness of each listener’s location. The reason the two data sets are split is to distinguish between the changing of the probe placement and the defined change in the carriers of the target devices.

The testing data set is what we perform the path reconstruction, pseudonym name changing scheme, and correlation identification between identifiers. While we show that these can be done for the initial data set, the new probe placement of the testing data set is better suited for the testing data set.
1.4 Contributions

This thesis contributes to previous theoretical tracking schemes that have been used (specifically in [44]). Before now, many of these tracking schemes have been based in simulations and do not consider a variety of environmental factors that apply to the detection of a target device. This thesis considers the environmental factors in detecting a path reconstruction and association across two commonplace protocols.

We also see the feasibility of de-anonymizing a simple pseudonym change scheme as packets are sent by the target device. This allows for a listener to associate an identity to a received address that may have changed since it has last been seen by the listener network.

Lastly, we shortly consider an implementation that uses the LTE protocol to track target devices across a Wide Area Network (WAN). This thesis shows that the precision necessary for tracking is not applicable to the environment of the placement of listeners, in addition to the limitations of off-the-shelf hardware in detecting responses to paging attacks.
Chapter 2

BACKGROUND

2.1 802.11 - WiFi

The 802.11 protocol is designed to be an effective communication protocol. For the remainder of this thesis, we will focus on the 2.4GHz spectrum and the applicable WiFi frames that are used in the context of the paper.

The 802.11 protocol splits the 2.4000-2.4835-GHz band into 11 channels, where each channel is spaced 5 MHz apart. The results in channels overlapping (except for channels 1, 6, and 11). Traditionally, most devices connect to these non-overlapping channels in order to filter out noise from other channels. Above the Physical Layer, 802.11 uses Carrier Sense Multiple Access (CSMA) in the MAC layer. This will prove to be useful as we will be able to see the Access point send out beacon frames signaling its presence in the area.

Above this defines the network layer. Each frame that is sent contains metadata for the packet, four addresses, and then the data to accompany the packet.

The four addresses in a WiFi frame are the Transmitter Address, Receiver Address, Source Address, and Destination address. Since this paper concerns targeting single devices, we look at either the source or the destination address. The other two addresses are can be different when transmitting data across access points. While the destination address does not guarantee that the target device has received the frame or is in the proximity of the access point, we can still determine that the target device was recently in the range of the transmitting access point.
In order to determine signal strength, a Received Signal Strength Indicator is a measurement that is taken at the client device in order to determine how well the connection a client has to the access point. This can be used as a rough indicator to tell how close the device is to the device that sent the packet. While this does not directly correlate to distance, this can be used as an indicator to where the device is in the environment relative to the listener. The relevancy of this indicator is discussed more in depth in the results section.

The scope of this paper is as a passive attacker that is not authenticated with the same network as the target device. Using ‘monitor’ mode, a passive attacker can capture metadata of WiFi frames that contains which devices are currently communicating within the spectrum, and other Quality of Service (QoS) information sent in management frames. Any information contained in the network section is not used in this study, and is typically not a reliable attack vector since this data can be encrypted.

The main feature of a WiFi network is the Network Discovery process. In this process, a client device must broadcast a probe request to find potential networks to connect to. Many implementations, such as the Unix command line tool ‘iwlist’, scan over each of the supported channels of the wireless interface to obtain a complete list of possible connection on the interface. The ‘iwlist’ source code can be found here [42]. Each network within range will respond with the corresponding SSID, frequency, and other general information about the network.

Note that the Network Discovery process can be arbitrarily used and some networks periodically broadcast their existence for possible clients to connect. For this thesis, we model a mobile target device that constantly needs to change networks based on the changing environment. As a result, the listener can monitor three common 2.4GHz channels (1, 6, and 11) in order to capture the network discovery process.
2.2 802.15.4 - Zigbee

Zigbee is traditionally deployed as a mesh network for Internet of Things (IoT) devices. This allows for easy routing of traffic throughout the network for low powered devices. In a Zigbee network, there are traditionally three roles that can be played by a device: a coordinator, a router, and an end device [20].

Zigbee has a lower data throughput than Bluetooth and Wifi, running at 250kbps in the 2.4GHz range [20]. Zigbee utilizes the ISM band in the 2.4 - 2.4835 GHz range, and has 16 channels (designated channels 11-26) each separated by 5MHz. Traditionally, Zigbee does not channel hop, but previous research has shown that channel hopping can be implemented in a network [18]. For the purposes of this thesis, the implementation omits channel hopping on the Zigbee network. Much like WiFi, Zigbee utilizes Carrier Sense Multiple Access (CSMA) in order to control when a device is available to transmit.

While the router and end devices are common in a larger Zigbee network, this thesis focuses on the coordinator device. The role of the coordinator is to forward traffic throughout the network, broadcast the existence of the network for new devices to connect, and handle the identifiers of the devices on the network. The main focus here is the broadcast of the existence of the network. Like a WiFi access point, a broadcast is periodically sent based on the specification of the network.

Each Zigbee network has an associated Personal Area Network (PAN) Identification, that allows other routers and end devices to identify and connect to the correct network. Many Zigbee coordinator devices scan the spectrum on boot. If there is another Zigbee network in its Personal Operating Space (POS), a random PAN Id must be chosen such that the new network PAN id is not in use by another network. As
a result, the newly formed network can broadcast to potential clients of the network. Once a new device is added to the network, the PAN Id is used to distinguish between network traffic that is sent on the same channel.

2.3 LTE

The Long Term Evolution (LTE) protocol is divided into over 70 bands with variable bandwidth and spacing. LTE bands range from the low side of the spectrum of 700MHz with carriers soon pushing out true 5GHz communication equipment. Some bands are reserved for specific communication, and many of the allocations are dependent on the region of the world. Within the United States, these bands are managed by the Federal Communications Commission (FCC).

LTE networks come in multiple forms. For the purposes of this thesis, we will focus on Orthogonal Frequency-Division Multiplexing (OFDM). In this application, a message is divided into multiple frequencies at a specific time frame. This allows for a higher throughput of data and negates the need for any guard bands in traditional Frequency Division Multiplexing.

Within the uplink and downlink channels, there exists the Physical Broadcast Channel (PBCH), Physical Downlink Control Channel (PDCCH), and Physical Uplink Control Channel (PUCCH), among others. Each of these are described in detail in the next few paragraphs.

The PBCH is the means of connecting to the network without any previous information about the bandwidth or the number of sub-carriers. This is strictly a downlink channel. The Master Information Block (MIB) gives information for decoding the rest of the sub-carriers in the PBCH as well as the bandwidth [15]. The most important
use of this channel for this thesis is detecting the presence of a User Equipment (UE) in the form of a paging attack. Paging attacks are discussed further in the related works and the LTE tracking section.

The PDCCH gives information to the UE about its allocations within the spectrum. The information sent in this channel is analogous to the management plane in a WiFi network. This contains any downlink resource assignments (such as bandwidth or sub-carrier), any uplink allocations for when the UE needs to respond, and uplink power commands [15]. For the purposes of this thesis, the information we want from the channel is the uplink allocation for the device. If we are able to obtain this information, then we can adapt to the correct Resource Allocation (RE) on the PUCCH and capture any data going from the UE to the eNodeB to determine the proximity of the device.

The PUCCH is a means for the UE to respond to the metadata that is sent in the PDCCH. Within this is a response to the eNodeB with the Channel State Information, Scheduling Requests, and acknowledgement of proper reception of data in from the eNodeB. While it would be ideal to capture any uplink data sent from the UE to the eNodeB, we focus on the the PUCCH because of the simplicity of the scheduling request. Without the implementation of our own LTE network, we must recognize the possibility that these uplink channels are encrypted and the response itself could be fragmented across multiple sub-carriers. The scope of this thesis is the detection of the target device and not the interpretation of the data sent from it. If we can detect the Scheduling request sent by the target UE, we define this to meet our goal of tracking the target device, as this implies the target UE is within the range of our listener. The application of these three channels for tracking are discussed in the LTE tracking section.
Chapter 3

RELATED WORKS

3.1 Jamming Attacks

Cognitive Radio development can be used to actively jam signals from a target once they have been identified. These jamming and anti-jamming techniques are discussed in [34]. As a counter to these jamming methods, Secure Encounters were introduced to allow for users to have more control over the exchange of their data [12].

There are several mitigation techniques to be considered that allow users more control. Further investigation into possible changes to underlying OS and networking stacks that allow the user to define when and where to probe for known networks without having to completely turn off certain communication features like WiFi or GSM. At a higher level, users should be able to enter known SSIDs and expected locations where they would appear. This would provide more of a restriction on probes for WiFi, but does not solve the problem of the privacy leaked by some other protocols commonly found in mobile devices today. While jamming attacks are not within the scope of this thesis, it is important to note that an application can essentially jam all communications going to and from a target if their location privacy is compromised.

3.2 Paging Attacks

Paging is done in an LTE network when a Mobile Management Entity (MME) needs to provide service to an end user. In order for this to be done, the UE must be found first, and then the data exchange needs to be set up. The first part of this is non-
trivial, since the MME does not know the precise location of the UE in its network, and there is a possibility that the device is no longer in the network. This is typically done by sending out a paging request on all eNodeBs within a radius relative to the last location of the device in the case of GSM [33]. Luckily, LTE allows for the idea of "smart paging" that sends out the broadcast only to the last eNodeB at which the device was seen, allowing for less traffic and a more precise location. This page is done on the PBCH, in which all communication is not encrypted [11]. Because of this, anyone with suitable equipment can decode the paging requests sent on the PBCH. This allows for a passive adversary to detect the presence of a device in the network.

Each UE needs to know when to respond to the paging request. Each UE is equipped with a pre-defined International Mobile Subscriber Identity (IMSI). This is unique to each UE within an area. With each paging request, either a Temporary Mobile Subscriber Identity (TMSI) or a Globally Unique Temporary Identity (GUTI) is sent to allow the UE to respond to a specific paging request. The LTE standard does not define when either the GUTI is changed, but it has been shown that the same GUTI can be used for multiple days before being reassigned [29]. We confirm this observation in the LTE tracking section and use this means to identify the presence of a target UE.

As mentioned before, this is one of the major vulnerabilities in the LTE communication protocol that leak information about connected devices. Namely, we look at paging attacks as a means to track a device in a network. This has been shown to be done over a larger area (2\(km^2\)) by a passive attacker in [40]. Shaik confirms that it is possible to actively track a device with the application of GPS and the Channel State Information, although this is outside the scope of this thesis. The attack vector that Shaik uses is based on assumptions that the target UE is equipped with
common social media applications (namely Facebook), and uses this as a means to trigger paging requests to the target UE.

The implementation discussed in this paper provides a basis for location tracking of a UE, but we aim to provide a fine-grained tracking application by tracking a response by the target UE by monitoring the PUCCH for a response by the UE. We will use a modified version of the paging attack used in [40] to trigger a response from the target UE.

3.3 Location Privacy

Previous work has been done in identifying what compromises location privacy of an individual. Since the explosion of mobile devices has happened, we have seen a growing concern for the privacy that an individual loses while carrying one around.

For the case of wireless LANs, the main concerns are the information leaked from time, location, sender identity, devices that receive corresponding packets, access point identity, and content [31]. We utilize multiple forms of this information (excluding the content of the data packets) to perform tracking in this thesis. Namely, the time, location that the packet was captured, and the associated destination and transmitter are essential information needed to associate identities to target devices. This paper mentions a pseudonym scheme to protect the identity of a user, but we show with a simple pseudonym scheme implemented, this alone is not an effective defense to protect location privacy within the WiFi protocol.

One proposed defense against this is to switch a pseudonym when a device enters a "mix-zone". A mix-zone simplifies as a designated location and/or time to switch a temporary identifier as a means to deter tracking. This can come in many forms
and traditionally is done by either switching identifiers with other devices in the same mix zone, or iterating through a pre-defined list used by the device. A game-theoretic analysis of the cost and benefit of the ideal time to change identifiers in a mix-zone is shown in [23]. We see that a device can still be tracked, even when switching its pseudonym in a mix-zone. We use some of the same mechanisms for identifying correlation between identifiers in this thesis.

3.4 Pseudonym Association Attack Model

The largest contribution and precursor to this thesis is described in [44]. Yieh formulates association matrices as a means to associate pseudonyms from one protocol to another.

Yieh uses two tracking vectors in order to implement his attack. The first of these is a long-distance pseudonym application named Dedicated Short Range Communication (DSRC). The purpose of this application is to allows a changing identifier to be seen throughout a tracking session. The second tracking vector modeled was WiFi, since this provides a constant identifier to match to the changing DSRC pseudonyms.

Yieh constructs two matrices in order to implement his tracking. The first of these is to determine associations between two DSRC pseudonyms. These are separated in categories based on the information obtained from the different tracking sessions. Based on pseudonym changes observed and Enter-to-Exit attack strategies, DSRC pseudonyms were associated to WiFi identifiers through a second constructed matrix. Through a series of transformations performed between the two matrices, Yieh successfully reconstructs paths taken by tracked pseudonyms and successfully de-anonymizes target DSRC and WiFi identifiers through simulation. The work done in this thesis takes the ideas presented in Yieh’s thesis and is translated to a real
world application to explore the implications this could have when applied to everyday devices.
We first attempted LTE tracking to monitor consumer cellular devices. The initial plan for this form of tracking was to allow for a long-range identifier to cover the entire area of the campus. Part of this process required identifying a tracked device’s temporary identifier known as the Globally Unique Temporary ID (GUTI) assigned by the Mobile Management Entity (MME). A simple overview of the device communications of the eNodeB, UE, and MME in an LTE network is in Figure 1. Once this is determined, the phone could be tracked in correspondence with it’s assigned downlink and uplink channel initiated by a Radio Resource Control (RRC) connection process.

Figure 1: Basic EnodeB, UE, and MME Connections

This chapter focuses on the implementation developed and feasibility of tracking a cellular device connected to an already established LTE network. Included is the idealized implementation, tools used to attempt tracking, and the implementation
building blocks that lead to our conclusion on the in-feasibility of tracking using an LTE vector.

4.1 Theoretical Implementation

In order to have a complete tracking vector through LTE there are a few that are necessary to know about the target device. Because the listener cannot be a middleman in the connection setup, the listener must be informed of the presence of the device on the local LTE network. As a consequence, the listener must know the carrier of the target device, as each carrier is assigned a multiple frequency bands within the LTE spectrum.

Once the frequency bands are narrowed down, the listener must scan each of these bands for the presence of the target UE’s corresponding Cell ID (that was established on connection to the network) in possible pages that are sent to the target from the network. Once a page to the target UE is found, the listener must monitor the RRC connection setup to determine the uplink channel that the UE will use to send data. This could theoretically give us a radius of the location of the target UE. This process is meant for only a single target device, and each target device must be tracked by each listener. Each step is a building block that would lead to the eventual tracking of the UE, as each process is highly dependent on the last.

4.2 SRS-LTE

SRS-LTE is an open source LTE platform in a software defined radio package that allows for the creation of custom LTE networks [3]. The implementation in this chapter leverages the cell_search application to search for nearby UEs and their corresponding
Cell IDs. While this identity is tied temporarily to a cellular device, this is neither the GUTI or IMSI that was discussed previously. The other vital piece of information obtained from this application is the uplink and downlink resource allocation for the device. This was used as a springboard to confirm that paging attacks could be launched against the target UE.

The other application used from SRS-LTE is the pdsch_ue application. Minor modifications were made to the application as discussed in the following "Paging Attacks" section. This application allowed for the monitoring of paging messages broadcast to the target device. This was used as both a confirmation that paging attacks were successful, and a measurement of the time taken to trigger a paging message. These results are discussed later in the chapter.

4.3 Paging Attack Implementation - A Building Block

In order to validate that a target device can be tracked, a simple paging attack was implemented in order to detect the presence of a LTE device. Before the paging attack was executed, it’s corresponding band and Cell ID was verified using the cell_search application from SRS-LTE. This allowed a monitoring of the appropriate downlink PDSCH channel to capture the P-RNTI message sent to the UE. Once this was verified, the target device was setup to receive push notifications. This was tested via two venues: Facebook Messenger and email push notifications. It has been shown previously that typing in Faceook Messenger to a target user is enough to trigger a paging occasion to its corresponding device [40].
4.4 Complete UE Tracking

One of the initial problems that deterred progress for this form of tracking is detection of an already-connected device. One of the simplest ways to perform this detection is to set up an IMSI catcher. This device acts as a middleman gateway connection to a true LTE network. As a part of the authentication process, a unique identifier known as the International Mobile Subscriber Identity (IMSI) is passed to the IMSI catcher. The authentication process is carried out by the IMSI catcher and then the GUTI assigned by the MME is passed by the IMSI catcher back to the now compromised device.

Because we want the listener to be a passive attacker, we do not want to interfere with the connection setup process. There needs to be a mechanism to detect a tracked device that is already authenticated with the LTE network which does not currently exist.

4.5 Experimentation Setup

As a means to use the LTE protocol as a tracking mechanism, this would require the use of triggering a P-RNTI to the User Equipment (UE) from the E-UTRAN Node B (eNodeB), then monitoring the RRC connection setup to move to the appropriate tracking channels (and frequency). The proposed experiment setup required recording the initial Cell ID via the cell_search application. Once verified, the pdsch_ue application started to monitor the PDSCH channel for paging requests to the corresponding cell.
4.6 Results

From the initial testing done on the LTE tracking vector, we determined that pursuing tracking via LTE was not possible in its proposed form. The limitations placed on the project were the limited resources for LTE platforms as an SDR package, as well as the inherent properties of LTE networks.

4.6.1 Using the SRS-LTE Platform for Tracking

The main issue with using the SRS-LTE platform is the flexibility of the products produced. While SRS-LTE allows for a fully-fledged LTE implementation, it is designed in such a way that is highly-dependent on the software defined radio that is being used. The documentation available is suitable for a simple implementation of an eNodeB and UE connection, while modification to the underlying frame access requires nitpicking from various places in the repository.

Thankfully, the sample binaries of cell_search and pdsch_ue provided a simple starting place for modifications. These provided a means to detect the UE and the any paging attacks that were meant for the UE.

While to original intention was to observe the continuous connection that a device made, this was not feasible given a low power and budget setup. The ability to monitor the resources assigned to a given tracking device would require an observation of the initial RRC transaction between the eNodeB and target device. Following this, both the common control channel, as well as the designated continuous communication channel must be monitored. Both channels must be monitored in the instance that the target device’s resource blocks are reassigned. While switching between the two channels is possible, SRS-LTE must reconfigure the SDR, this process takes on the
magnitude of seconds, and has a high probability of missing key radio frames that are on the order of 10 msec per frame. While this was a major hindrance, we still performed initial testing with paging attacks and considered other channels to perform our tracking as discussed in the following subsections.

4.6.2 Paging Attacks: Successes and Failures

Paging attacks were initially a success. While we were able to elicit multiple successful paging attacks against a target UE that was placed in close vicinity to the SDR. The UE band was predetermined and both Facebook Messenger and email were used to create a successful paging attack [21].

While these attacks were carried out successfully, we tried to initiate a paging attack against a device that was not currently connected to the network, but had recently been connected. Initial testing was done with varying time after disconnection. The longest observed time between a UE disconnection and then a page sent to this disconnected UE was 35 seconds past the disconnection time. This is a significant amount of time that the UE has left the network and was less than desirable for tracking.

4.6.3 Lack of LTE Network Control and Information

One of the largest hindrances to using an LTE network as a tracking vector is the coverage area of a given LTE network doesn’t allow for precise tracking of the target device.

As mentioned earlier, we cannot easily monitor the uplink channel. We could settle for monitoring the shared downlink channel used to distribute RRC requests. While
we have seen that these paging attacks can trigger a request to a tracking device, we see a single response from the target device (if it is within the coverage area) as a part of the RRC connection process. Once the connection is set up, it was observed that the continuous communication channel is encrypted. While the focus does not include the actual data transferred, the underlying communication structure of SRS-LTE does not support following the established encrypted data stream. It is out of the scope of this thesis to implement an LTE application that allows following this data stream due to immense measure of the current LTE specification.

Taking a further step up, if only the PDCCH and PUSCH channels were monitored for the exchange of P-RNTI and the initiation of a RRC: Paging requests, we would know the destination UE, but no way of knowing where in the LTE coverage area this device is currently being used. This proved to be an order of magnitude too large to be used as a tracking vector given the proposed environment of the Cal Poly campus.

4.7 Conclusion

While LTE can be used as a tracking vector in a large scale deployment, the results obtained did not prove to be useful for the experiment in its decided form. The scale of tracking was an order of magnitude larger than was desired for the experiment, and the hardware necessary for a complete listener implementation did not meet predetermined scope of the project.
Moving from using LTE as a tracking vector, we discuss associating anonymized Zigbee devices to their corresponding WiFi devices in this chapter. In order to associate pseudonyms to their corresponding WiFi addresses, we use apply the following attack model as seen in Figure 2 which is based off of the attack model proposed in Yieh’s thesis [44]. Modifications to the attack model were made to accompany the Movement Matrix Restrictions, and to use the Zigbee packet captures entry data rather than the long-distance protocol used by Yieh. All subroutines are run and information flows in the direction of the arrows. The final product is the W2Z Matrix, which can then be used for path reconstruction once a confidence threshold is set. Each tool is specified in the following subsections.
5.1 Data Matrices

In order to make associations, we use two different matrices as our main data structures: A WiFi to Zigbee (W2Z) matrix, and a Zigbee to Zigbee (Z2Z) matrix.

The Z2Z matrix is used to make associations between two different Zigbee pseudonyms $z_i$ and $z_j$. The possible outcomes of associations are denoted as associated, disassociated, possibly associated, and not enough information to conclude a type of associa-
tion. In order to come to these conclusions we first find disassociations, perform the same listener attack followed by the exit-to-enter attack, and lastly apply a transitive property over the matrix. Each of these are explained in following subsections.

The W2Z matrix is used to make associations between the target WiFi addresses and the Zigbee Pseudonyms that were seen during the tracking process. An example matrix is pictured in Figure 4. Here we start with each Zigbee pseudonym equally likely to be associated to a WiFi address. Note that the columns all sum to 1, with a value of 1 meaning an association, 0 meaning a disassociation, and any value between the two are denoted as a possible association.

The W2Z matrix is manipulated after the Z2Z modifications are made according to the attack model. The W2Z matrix is initialized with an equal chance of Zigbee pseudonym $z_i$ associated to each WiFi address $w_1$ through $w_n$ for $n$ WiFi target devices. First, WiFi to Zigbee disassociations are found, followed by finding known associations. Finally, the boosting process is applied, and any entry $e_{ij}$ in the W2Z
matrix with a probability of association above a predetermined confidence threshold is marked. The variation in confidence threshold is examined in the results section.

![Figure 4: Initial WiFi to Zigbee Matrix](image)

### 5.2 Association and Disassociation Classifications

There are three types of associations that are clarified in this section. These concern known associations, possible associations, and disassociations.

Known associations are made between a WiFi address $w_i$ and a Zigbee pseudonym $z_j$. Known association are when the WiFi address $w_i$ and Zigbee pseudonym $d_j$ are seen together and there are no other pseudonyms seen during that time. While it is possible to lead to a wrong known association made, it is highly unlikely.
An instance when this could happen is when a Zigbee broadcast is sent from device $d_x$ and leaves before a WiFi broadcast is sent. Within a small time frame at the same listener, a WiFi broadcast is sent from device $d_y$ and leaves before a Zigbee broadcast is sent. In most instances, both tracking vectors are observed, and the observation period moves in to the possible associations category.

Possible Associations occur when there is are multiple WiFi addresses seen at a listener with one or more Zigbee Pseudonyms. This occurrence does not allow for the attacker to make a definitive association, but allows the attacker to leverage this information for the boosting process.
Finally, a known disassociation is between a WiFi address $w_i$ and a Zigbee pseudonym $z_j$. If $w_i$ is seen at listener $A$ and $z_j$ is seen at the same time at listener $B$, then we know that $z_j$ cannot be associated with $w_i$.

![Figure 7: Example Known Disassociation](image)

### 5.3 Exit-to-Enter Attack

The Exit-to-Enter attack is used to find associations and possible associations for the Z2Z matrix. When a target device $d_i$ enters the coverage area of a listener, the first packet received from $d_i$ by the listener is the start of the observation period and the end of the observation period is 20 seconds after the last packet is seen from the $d_i$. This excess time after the last packet is to distinguish between two different events. The longest heartbeat rate of our communication protocols is 15 seconds, so a value slightly longer than that is used to separate observation periods from each other. This is particularly useful if $d_i$ leaves the coverage area of a listener, and then immediately returns to the same listener. These are broken in to two distinct observation periods to be linked by the Exit-to-Enter attack.
The Exit-to-Enter attack uses the first observed Zigbee packet in an observation period as the "enter" event, and the last time the corresponding Zigbee pseudonyms is seen at the listener as the "exit" event. A notion of these events are given in Figure 8. Note that it is possible for the enter and the exit events to be from the same packet, as we have no notion of the physical location of the target device within the coverage area.

After splitting the set of Zigbee packets into two sets of all enter events $E$ and a set of all exit events $X$, we construct a bipartite graph that maps from each exit event to each enter event. We then use Python’s NumPy package to solve for a minimum weight perfect matching on the bipartite graph [32]. An example construction of the bipartite matching problem is in Figure 9.
Figure 9: Exit-to-Enter Bipartite Matching

In order to assign weights to each edge, Yieh uses a “training” model to assign weights in the bipartite graph based on the expected behavior of devices when traveling between listeners [44]. Because Yieh uses a simulation, he uses non-anonymized data to record the number of devices that traveled from listener $A$ to listener $B$ and the total travel time taken, then calculates the average travel time $t_{avgAB}$ between listener $A$ and $B$. This is calculated for each pair of listeners. While Yieh uses de-anonymized data, we explore the results of using de-anonymized data vs anonymized data during the training aspect of the Enter-to-Exit attack in the results section.

Each edge in the bipartite graph is weighted according to the type of associations between the events. For any known associations, a large minimum value is assigned to ensure that the edge is chosen between the exit event and the enter event. For any known disassociations, the weight of the edge is assigned as a small positive value, which will not be picked by the algorithm. For the rest of the weights, we use the training model to determine the appropriate weighting. The training model is applied
to the edge where the exit event is at listener $A$ and the enter event is at listener $B$. The difference in time between the exit and enter event $t_{\text{delta}}$ are computed, then compared to the average time taken to travel $t_0$. $t_{\text{error}}$ is defined as the percent difference between the actual travel time and the average. $t_{\text{error}}$ is calculated as either $2 - \frac{t_{\text{delta}}}{t_0}$ if $t_{\text{delta}}$ is less than $t_0$, or as $\frac{t_{\text{delta}}}{t_0}$ if $t_{\text{delta}}$ is greater than or equal to $t_0$ as in Yieh’s paper [44]. Finally, the weighting is applied as

$$- \frac{n_0}{t_{\text{error}}}$$

which correlates the number of times the path between the exit listener $A$ to enter listener $B$ was taken in relation to the error of the time actually taken to travel between the exit listener $A$ to entry listener $B$ [44].

The output set of edges chosen are then compared to the maximal weighted edge in the graph, and if above a predetermined threshold percentage of the maximally weighted edge, the exit pseudonym $z_i$ is denoted as associated to the linked enter pseudonym $z_j$. For any other edges in the solution, the linked pseudonyms are marked as possibly associated. The overall computation of this problem is heavy, as the underlying minimum weight perfect matching is a combinatorial optimization problem, which leads to long run times for larger sets of observation intervals.

5.4 Transitive Property

At the end of the Z2Z matrix construction, the transitive property is applied over the matrix. Namely, if Zigbee pseudonym $z_n$ is known to be associated with $z_m$, and $z_m$ is known to be associated with $z_p$, then we know that $z_n$ is associated with $z_p$. This follows the same precedence for transitive disassociations. We have confirmed Yieh’s
findings that using the transitive property over possible associations causes a cascade of incorrect associations [44].

5.5 Boosting Processes

In a process that Yieh dubs as “boosting”, the context in which each Zigbee pseudonym $z_i$ is seen with each WiFi address $w_j$ is taken into consideration [44]. Boosting is introduced as a way to shift the probability of association from within a column to Zigbee pseudonyms that have been seen more times with a given WiFi address.

The formula followed for redistribution is depicted in Figure 10. Yieh applies three levels of boosting, where each level has a diminishing impact on the outcome of the redistribution. In this thesis, we utilize two, as the third level had little impact on the outcome of our associations, and only increased the rate of false positives during association. From the figure, we can see that $z_1$ has been seen with WiFi identifiers $w_1$ and $w_2$. Thus we redistribute using the delta value to take a portion of $v_3$ and $v_4$ and apply it to $v_1$ and $v_2$. The portion that is redistributed is denoted as $\Sigma$. The value $c_1$ and $c_2$ are derived from the number of times each identifier $z_1$ was seen with. This applied a form of a weighted average, where $c_1 \leq c_2$ if $z_1$ was seen more with $w_1$ than with $w_2$. This form of redistribution is done column-wise, and maintains that each column sums to one.
There are two levels of boosting used, each to reference different contexts in which associations can be made. These are each labeled in their respective levels of 1 and 2. Each follows the same methodology as Yieh [44]. Note that Yieh uses 3 levels of boosting, where our testing led to a significant jump in false positives when the third level of boosting was applied. Level 1 boosting is done for any immediate possible associations, and level 2 for a single-degree of separation according to the Z2Z matrix. The main variable tested in the results section is Yieh’s Δ value used for redistribution in his calculations. The Δ value determines the portion to be redistributed over the column as seen in Figure 10. The impact of this variable is analysed in the results section.
5.6 Confidence Threshold

Once the W2Z matrix is completed, then a predetermined confidence threshold is used to determine which possible associations should be marked as actual associations. While the results section has an analysis of a variable confidence threshold, it is set at 0.5 unless otherwise specified. This way if the entry $e_{ij}$ in the W2Z matrix for WiFi address $w_i$ and Zigbee pseudonym $z_j$, then $w_i$ is deemed associated to $z_j$.

5.7 Path Reconstruction

Using the results of setting the confidence threshold for the W2Z matrix, we can reconstruct pathways given the set of Zigbee pseudonyms that are associated to the given WiFi address. Further analysis and examples are given in the results chapter.
MULTI-VECTOR LISTENER IMPLEMENTATION

In this chapter we discuss the implementation of tracking devices that allow for the collection of data which we define as listeners. Each listener was placed around the Cal Poly campus in highly trafficked areas. These listeners were equipped with an 802.11 and 802.15.4 interface in order to capture data for both tracking vectors. We cover the implementation for each interface and the privacy concerns that come with capturing data in a public environment.

6.1 802.11 Tracking

The listener implementation utilizes the Nexmon firmware patch to monitor WiFi traffic destined to tracked devices [39]. Monitor mode and its capabilities are explained in Chapter 2. Each listener is run on of a Raspberry Pi 3B+, which is equipped with the Bcm43455c0 Broadcom wifi chip [1]. In our implementation, we use The Nexmon firmware patching framework that allows for custom input-output control (IOCTL) implementations to allow for further control of the chip. The original implementation allows for the user to send an IOCTL command to put the device into 'monitor' mode, which was not previously possible through a pure software interface on the Raspberry Pi. Monitor mode allows for a user to observe packets that are captured on an interface that are not destined for the device that has captured the packet.

The 802.11 tracking implementation utilizes the Nexmon firmware patching framework in addition to the iwconfig utility to monitor the state and change channels on the monitoring interface [4, 39].
There is an issue with monitor mode periodically falling into a firmware trap [5]. When this happens, the chipset stops responding to any inputs and is in an unusable state. In order to combat the possibility of the firmware failing, the system periodically checks the current channel. If the response from the framework is a failure, the firmware is unloaded with a call to modprobe, and then reloaded. While this has been noted to solve some issues, the reload could still fail. If this is the case, the probe logs the failure and the time, then is automatically rebooted.

6.2 802.15.4 Tracking

This project applied Dresden Elektronik’s ZSHARK firmware and tooling [7]. This is typically used for debugging of existing Zigbee networks, but can be set as a third party listening to a chosen Zigbee channel. There is no need for the device to be actually authenticated to the existing network, unless the data is encrypted. Since the scope of this thesis is only detection of a target device, this is not necessary, the only knowledge an attacker needs ahead of time is the channel the target device will use. While certain implementations of 802.15.4 implement channel hopping, for testing we set the target device to adhere to a single channel.

A Dresden Elektronik Wireless RaspBee Premium ZigBee addon is attached to the GPIO pins on the the listener [6]. The ZSHARK firmware must be loaded once, and then all subsequent power cycles load the correct firmware. On boot, the ZSHARK tool is started, set to the designated tracking channel and a filter is set to only collect the target device 802.15.4 MAC addresses. This is then piped to a PCAP which is stored and analyzed offline. The tool runs for the duration of the tracking period and a separate PCAP designated on every boot.
6.3 Controlled Tracking Experimentation Setup

In this project, we did not want to collect data from any device that has not consented to being tracked. In order to not infringe upon the privacy of other students on campus, each listener must filter traffic that is explicitly a part of this thesis.

For the listener implementation all traffic was sniffed, but only the devices that had consented to being tracked were stored. While tracking, each interface contained a pre-determined filter of the MAC addresses of the consenting devices that was loaded on boot. As packets were received on the interface, the filter would only allow packets that passed the filter to be stored.
MULTI-VECTOR TARGET DEVICE IMPLEMENTATION

In this chapter we discuss the implementation of each device that was tracked. Each target device had a unique MAC address in both the WiFi and Zigbee vectors, but for all other intents and purposes were flashed using the same images and hardware. This would ensure to limit the same behavior between target devices. On boot, both the Zigbee and WiFi interfaces automatically started via initialization scripts. A picture of the hardware is included in Figure 11.

![Target Device Implementation](image)

Figure 11: Target Device Implementation
7.1 802.11 Tracking

Wireless presence of the target device was made known by the periodic probing of available access points in its vicinity. While access points periodically have beacon frames, we utilize the fact that many end devices can send broadcast frames to find available access points. A response from an access point is not necessary for tracking, although any responses received can be used to extend our tracking range leveraging the ”hidden terminal” problem as seen in Figure 12.

![Figure 12: Leveraging the Hidden Terminal Problem](image)

A simple python script was designated to launch once the RPi was booted. This designated the creation of a broadcast frame sent over the current WiFi channel, and then a channel change to either 1, 6, or 11 with equal probability every 5 seconds. It was possible for RPi to not change channels in between frames sent if the result of the random generator resulted in the current channel of the device.
7.2 802.15.4 Tracking Implementation

deCONZ is software created by Dresden Elektronik that acts as an 802.15.4 network manager and gateway [2]. This software was designed specifically for the RaspBee Premium ZigBee addon for the RPi [6]. The highlight of this software is that each device can be set as a gateway and broadcast the existence of its network to non-authenticated devices. This is utilized to designate a separate network for each target device, which broadcasts the existence of its network to be captured by the listener.

On boot, the RPi is set to launch the deCONZ software. The designated tracking channel is set, with no other network configurations made, and the network is launched with the target device as the gateway. Every 15 seconds, a beacon frame is sent to broadcast the availability of the target device’s network. Included in this broadcast is the MAC address of the target device. This program is run for the duration of the tracking period, and is automatically restarted on reboot.
In this chapter, we discuss the experiment environment setup and procedure in detail. This includes a description of the target device setup and carriers, listener placement, expected listener coverage, and the tracking network operation timeline.

8.1 Environment Overview

This experiment was developed and carried out on the California Polytechnic University: San Luis Obispo campus [10]. This was chosen as a proper environment as there is already wireless network infrastructure for WiFi used by students and various Bluetooth and Zigbee devices throughout campus. Because of the existing infrastructure, our tracking devices could receive wireless traffic inbound to the devices, rather than just outbound traffic, thus increasing our tracking area through the already existing network as explained by the modified hidden terminal problem in Figure 12. No changes to the existing infrastructure were made to accommodate this experiment. Each listener was temporarily connected to the wireless network on boot to update time, then disconnected once the process was finished. This allowed for easier tracking of boot times and logging any failures. Besides this instance, the listener had no other interactions with the wireless infrastructure.
8.2 Listener Setup and Placement

Listeners were all placed in areas that were available 24 hours a day and all probes except one were exposed to the outdoor elements. This required the electronics to be in a waterproof container and locked up. The waterproofing was not tested extensively before the project, but every listener was fully functional despite the multiple rainy days through the experiment. A picture of the waterproof container and experiment notice is included in Figure 13. The experiment notice was placed on top of the box in order to inform passerby that the opaque container was part of a student experiment and was not to be touched or moved for data integrity purposes.

![Figure 13: Listener Container and Protection](image)

Each listener placed included a Raspberry Pi 3B+ with a 20800 mAh battery with a 32 GB SD card to collect all incoming packets for the tracked devices. The Raspberry Pi was fitted with a Dresden Elektronik Wireless RaspBee Premium GPIO module to collect corresponding Zigbee traffic [6]. These were enclosed in an Apache Weath-
erproof case which was then cable-locked to the closest sturdy object (e.g. Tree, Bike Rack, Stop Sign) with a padlock. The setup is depicted in Figure 14. The board orientation within the case was controlled to always have the Zigbee module face up with the antenna facing away from the battery.

![Listener Implementation](image)

**Figure 14: Listener Implementation**

This project was broken in to two phases: Test placement and Data Collection Placement. This was necessary to separate as the listener placement was changed between the two sets of data collected. The test placement was used over the course of two weeks to determine if the volume of packets was adequate for tracking. Once this time frame was finished, analysis showed two probes did not capture sufficient traffic in order to reconstruct pathways. This is explained in detail in the results section.
The data collection placement was the placement used to analyze the effectiveness of the pseudonym tracking scheme.

### 8.2.1 Test Placement

The initial placement of the devices were intended to capture all traffic that comes and goes from the three main entrances to the Cal Poly campus. The three entrances covered were California Boulevard, Foothill Drive, and Grand Avenue. The intention was to document each device being tracked with their entrance and exit of the campus. This would give a time reference to which devices were valid to track for the day if a volunteer did not have a reason to come to the campus on a given day. The placement of each listener for this phase is in Figure 15.

![Figure 15: Expected Listener WiFi Coverage for Testing](image)

Following the test placement, one probe was moved. The probe placed to capture Grand Avenue traffic was moved next to the Computer Science Faculty offices. While
this provided overlap from the probe that was placed outside of the Frank E. Pilling building, this covered another high-traffic area while allowing a more precise path reconstruction when analyzing packet captures. The placement of each listener for this phase is in Figure 17.

8.3 Data Collection Listener Coverage

Each listener has a corresponding coverage area. This was mapped during low foot-traffic times on campus. Each probe was in its designated tracking location. Provided in this section is a comparison between the expected coverage area and the coverage mapped for each of the probe placements. This is important to note that

8.3.1 Expected Coverage Measurements

An open field was used for testing in order to minimize multi-path effects that could occur. The data was collected for each probe is as follows. For each listener, the device was booted, the time was updated, then a timer was set for 10 minutes. A target device was then activated to send out a Zigbee probe every 15 seconds and a WiFi Probe every 5 seconds. This allowed for the collection of 40 Zigbee packets and 120 WiFi packets for a test section.

A straight line was followed extending from the enclosed listener out to the open field. Initial testing provided a reference close to 200 feet for Zigbee and 170 feet for 2.4 GHz WiFi as packet loss references. While walking out toward the reference point, GPS references were taken every 5 seconds. Once a test was completed, the enclosure was rotated 90 degrees in order to determine if the rectangular shape would provide an abnormal coverage area.
Following the initial testing, the same procedure was executed, except this time a circular path was taken close to the drop line in order to confirm that the shape of the coverage was mostly circular. Note that each GPS coordinate comes with an error of at least +/- 8 meters due to the inaccuracies that are inherent in GPS. Figure 16 is an example recording of GPS coordinates overlaid on a Google Maps to determine the furthest distance a probe packet was received [25]. Note that not every probe packet was matched with a GPS reference because some packets were outside the reception range of the listener.

![Figure 16: Expected Coverage of Zigbee Measurements from GPS Coordinates](image)

8.3.2 Actual Coverage Measurements

Information from the expected coverage measurements were used to determine an appropriate pathway to cover when testing in-place measurements. The measured coverage of devices are shown in Figures 17 and 19. While these figures show the results expected if no structures interfered with communications, this is not a realistic expectation. Figures 18 and 20 show the reception of each of these beacons. Structure interference and multi-path effects are two effects that cover the discrepancies between the expected and measured coverage of each listener.
Figure 17: Expected Listener WiFi Coverage for Data Collection

Figure 18: Measured Listener WiFi Coverage for Data Collection
Figure 19: Expected Listener Zigbee Coverage for Data Collection

Figure 20: Measured Listener Zigbee Coverage for Data Collection
Note that there are overlapping sections for two pairs of beacons. While this is not an instance that Yieh covers in his thesis, this provides a more fine-grained location of a device if seen by two listeners at one time. This is covered in more detail in the results chapter.

8.4 Listener Operational Period

The tracking network implemented was intended to collect each packet that Each probe was turned on in the morning between 5:45-6:30 AM. This was required in order to make sure that all the traffic for the target devices was captured for the day. Each day the devices were collected between 7:00-8:30 PM. Between the boot and power off processes was considered ”core hours” for students, and thus time to track devices. Devices were not tracked on the Saturdays and Sundays of each week since no classes were offered on campus at this time.

8.5 Target Device Setup

Each target device was equipped with a 20800 mAh battery and a Raspberry Pi 3B+ equipped with a Dresden Elektronik Wireless RaspBee Premium GPIO module. When powered on, the Raspberry Pi would boot, the Zigbee gateway operated on the RaspBee module would initialize shortly after the WiFi interface was started. The Zigbee gateway was then set to a heartbeat of 15 seconds. The WiFi interface was designated a heartbeat of 5 seconds. If either the wireless or zigbee interface was unable to beacon, the device would be power cycled.
8.6 Target Device Carriers

Each carrier of a target device was a student at California Polytechnic State University: San Luis Obispo. Each student was in a major under the College of Engineering, with four graduate students and one undergraduate student. Students were made aware of the number of listeners that were tracking the devices that they carried, but were not told their locations. Because each listener location was in a public place, it is possible that the devices were discovered as the carriers proceeded with their schedules, but any discovery of the listeners was not logged as a part of the experiment.

8.7 Data Collection and Project Privacy

Since this experiment was run on public university campus, proper precautions were taken to ensure the privacy and protection of students. The University Police Department was made aware of the project, and the time frame that the listeners would be collecting data. In addition, each listener was labeled distinctly as a student project with the contact information for the project creator, the project advisor, and the computer science department in case of questions or problems.

The main concern when collecting data was to limit the packets collected to those only pertaining to the experiment. Any packets that were addressed to or sent from the target devices were captured. Since the tracked devices were never connected to the wireless infrastructure already available on campus, it is highly unlikely that any devices other than available access points responded to the network broadcasts sent by the tracking devices. Since the devices acted as Zigbee Gateways, they were discover-
able, but were set up to not allow for any outside Zigbee devices to be authenticated with the gateway.
Chapter 9

PSEUDONYM APPLICATION

In this chapter, we document the pseudonym implementation that was overlaid on the collected data in order to determine the feasibility of tracking a multi-vector pseudonym switching devices. The real world attack implementation required developing a restriction based on human movement between the listeners that were placed. Adaptations were made from Yieh’s implementation to account for discrepancies between a real world implementation and simulation.

9.1 Protocol Choice and Pseudonym Application

Previous work has used DSRC as a long-distance pseudonym changing radio protocol and WiFi as a short-distance non-changing signal protocol in order to perform their tracking analysis [44]. In this experiment, WiFi is still used as the short-distance non-changing signal protocol, although DSRC is replaced by Zigbee. Zigbee acts as a long-distance protocol, although pseudonyms have been applied to periodically change the target device’s Zigbee MAC address to obtain the same pseudonym behavior. Before, DSRC provided a coverage area that was 2000 meters in diameter. This was more than a magnitude of area larger than our coverage area for Zigbee of a 400 foot diameter coverage area. This difference in magnitude leads to a smaller gap between our long-distance and short-distance coverage areas.
9.2 Pseudonym Switching Scheme

The implemented pseudonym switching scheme employs a simple counter to the Medium Access Control (MAC) address assigned to the device. The data collection period was over the process of 3 weeks. In order to preserve the non-overlapping identifiers needed for unique pseudonyms in Yieh’s paper, there needs to be a restriction for the pseudonym change frequency. In order for these to be unique using the simple counter we use the smallest numerical difference in two of the tracking device’s MAC addresses. In this experiment, the closest were the MAC addresses had an available gap of 11012 addresses before the counter would overlap. Over the course of 23,501 minutes, the pseudonym could be changed at most every 128 seconds without intersecting another set’s address space. The collected data has been tested using varying pseudonym switching rates in order to measure the impact of switching frequency on privacy preservation in the results section.

Algorithm 1 displays the process to apply pseudonyms to the already captured data. We start with the beginning of the tracking period $t_0$, all packets $pkts$ and a predetermined pseudonym switching rate $r$. For each packet, we modify the address based on the time since the beginning of tracking $t_0$ and the rate at which the address should change $r$. Once all the packets in this range have been modified, the counter
is updated and the new range of packets are modified. This continues until all packets have been modified to their new pseudonyms.

**Input:** \( t_0 \): Beginning of tracking period, \( t_{end} \): End of tracking period, \( r \): rate at which to apply a new pseudonym, \( pkts \): list of all captured packets in time order

**Result:** \( pkts_{pseudo} \): Anonymized list of packets

\[
\text{counter} \leftarrow 0 \\
\text{beginApp} \leftarrow t_0 \\
\text{endApp} \leftarrow t_0 + r
\]

\[
\text{while } \text{beginApp} < t_{end} \text{ do} \\
\quad \text{foreach packet } p_i \text{ where } \text{beginApp} < \text{time } p_i \text{ received} \leq \text{endApp} \text{ do} \\
\quad \\
\quad \quad p_i \text{ address} \leftarrow p_i \text{ address} + \text{counter} \\
\quad \end{align*}

\[
\text{counter} \leftarrow \text{counter} + 1 \\
\text{beginApp} \leftarrow \text{endApp} \\
\text{endApp} \leftarrow \text{endApp} + r
\]

\[
\text{end}
\]

**Algorithm 1:** Pseudonym Application overlays the non-anonymized data with a pseudonym based on the switching rate and time since tracking has started

### 9.3 Movement Matrix Restriction

One of the applied attacks that results in known disassociations leverages a global view of the tracking network. If a pseudonym \( p_1 \) is seen at listener A, and pseudonym \( p_2 \) is seen at listener B at the same time, we know \( p_1 \) and \( p_2 \) are disassociated.

We propose and implement a matrix that filters the possibility of a device being seen at listener A, followed by a predetermined time delta depending on the distance to
another listener B. The purpose of this matrix is establish disassociations based on the ability for a target device to move between locations in the predetermined time delta. The time taken to travel between listeners is based upon the shortest path that can be taken and is measured at a walking pace and is meant to be a lower bound on the time it would take to travel between listeners. This is meant as a lower bound as a target device could take a non-direct path between listeners.

Figure 21 shows two different pathways originating from the listener near the Frank E. Pilling on the Cal Poly campus. We can see the path from the Pilling building to the Kennedy Library is shorter than the path from the Pilling building to the H4 Parking Lot. We assign the entry in the Movement Matrix from the Pilling listener to Kennedy Library as 20 seconds and

![Figure 21: Movement Matrix Adaptation](image)

This is a symmetric matrix, as it is assumed that moving from Listener A to Listener B will take the same time as moving from Listener B to Listener A.
Chapter 10

RESULTS

Using the experimental setup and the data collected from the three week period as explained in previous chapters, this chapter documents the impact of varying inputs such as association confidence threshold values, pseudonym change rate, and training variation for the exit-to-enter attack strategy.

There are two measurements used throughout this chapter to compare the outcomes of variation. Their definitions are in equations 9.1 and 9.2 and match the analysis used by Yieh. Precision determines the how many correct associations were made compared to the total guessed associations. Recall is a measure of correct actual associations made out of all associations that could have been correct. These measurements reflect the effectiveness of the pseudonym associations made.

\[
Precision = \frac{TruePositive}{TruePositive + FalsePositive} = \frac{Correct\ Associations\ Made}{Total\ Guessed\ Associations}
\]

\[
Recall = \frac{TruePositive}{TruePositive + TrueNegatives} = \frac{Correct\ Associations\ Made}{Total\ Actual\ Associations}
\]

We utilize the NumPy and Pandas from the SciPy library to implement the tracking algorithm and tune the variables throughout the analysis [32].
10.1 Varying Pseudonym-Switching Rates

The rate at which the pseudonym changes is one of the factors that can deter the effectiveness of the proposed tracking technique. The switching rates were tested in increments of 5 minutes starting at changes every 5 minutes up to every 120 minutes. The total pseudonyms encountered for each switching rate is seen in Figure 22. We fix a $\Delta$ value of 0.6, the confidence threshold at 0.5 and apply the attack model to each of the pseudonym switching rates.

![Total Pseudonyms Seen vs. Change Rate](image)

**Figure 22: Total Pseudonym Counts for Corresponding Switching Rate**

Figure 23 sees a general decrease in precision and recall as the switching rate increases. Nominally, we have a range from 5.9% at the faster rates of change, up to 11.03% for recall. In terms of precision, we achieve a fairly comparable result of just above 65% at the faster rates of change, up to an 85.2% precision. We can attribute this to the larger Z2Z matrix constriction, as well as the increased number of observation intervals that exist when applying the Exit-to-Enter attack. Note that Yieh's precision percentage
was on average 78% and recall on average 17%. This gives us a reference to the effectiveness of a real-world implementation. Our results are comparable in terms of precision, but our recall falls short. We can attribute the lower precision and recall rates to various factors. Namely, we have a much smaller coverage area than that of Yieh’s simulated listeners and the number of tracked devices is significantly smaller. This inhibits the ability to collect more information for our Exit-to-Enter strategy, the Same Listener Attack, and known associations. There are also several instances in the data set where we see a Zigbee pseudonym, but no way to associate to a WiFi address.

![W2Z Precision and Recall vs. Change Rate](image)

**Figure 23: W2D Association Success vs. Pseudonym-Switching Rate**

### 10.2 Varying Confidence Threshold

One consideration when applying associations between pseudonyms is the association threshold. We analyze the final confidence threshold as a means of determining the trade-offs of allowing through possible false positives in our final W2Z association set used for path reconstruction. Consider an association in the W2Z matrix that is just
above the 50% confidence threshold. While this meets the set confidence threshold, we would much rather choose an association that is much higher, representing a more-likely association. As a trade off, we would see less associations pass the confidence threshold, but the attacker would be more sure of their tracking methods.

Figure 24 displays the effect of varying the confidence threshold for the W2Z matrix ranging from 0.5 to 0.65 in 0.05 percent increments. We set the delta value to 0.6 and the pseudonym change rate to 300 seconds. The training for the Exit-to-Enter attack used de-anonymized training data. We can see the general trend discussed above. As the confidence threshold increases, there are significantly less associations made, but the recall is considered higher in these cases. In the other direction, a lower confidence threshold allows for more associations to be made, but less of them are correct. This meets our expectations that our attack model is behaving correctly. Machine learning could be used in order to find the optimal confidence threshold that an attacker could use to make the most of their resources. If an attacker had more than enough resources, they could afford to chase false positives in their model as a means of covering possible associations that could be their target.
10.3 Delta Tuning for Redistribution Analysis

The boosting process requires an input value for $\Delta$, which determines the amount of probability redistributed during the final phases of the W2Z matrix construction phase. Yieh uses a common value of 0.6 for $\Delta$, and explains that trial and error led to the choice [44]. We aim to provide more insight to the $\Delta$ variable in this section.
Figure 25: Precision and Recall Results for Varying Delta

Figure 25 shows the results of running tests using a varying $\Delta$ value that ranges from 0.4 to 0.8 in 0.2 increments. This is run with the varying pseudonym changing rate to the relationship between the two variables. From the figure, we can see that as delta increases for a given pseudonym change rate, the general trend is a decrease in precision and a general increase in recall.

The main reason we see the precision decrease and the recall increase as we increase delta matches the analysis by Yieh [44]. As we increase delta, we see a higher number of W2D associations made, but this leads to many incorrect associations. The effect is seen during the level 1 boosting stage, where redistribution overcompensates for the number of times that a Zigbee pseudonym is seen with a given WiFi address. If the wrong association is made at the level 1 boosting stage, it is difficult for following levels of boosting to correct the error made. To follow this, as delta decreases, we are more careful about which Zigbee pseudonyms are considered associated, and because of this, we see the recall increase.
10.4 Association Training on Non-Anonymized Data

The association training stage in Yieh’s tracking algorithm has an idealized implementation that relies on tracing each pseudonym back to its target device [44]. This allows for a more accurate depiction of how many vehicles traveled between each listener. When the data is anonymized, there are theoretically more vehicles and

In this section, we fix a delta value and a confidence threshold for considered association. For the first run of the tracking algorithm, we use the non-anonymized data in order to train the expected time between listeners. This is the idealized situation as we know exactly how many vehicles traveled between each node and their corresponding times.

![Association Training Data Precision vs. Change Rate](image)

**Figure 26: Exit-to-Enter Training Phase Variation Precision**
For comparison, we use the anonymized data as a training set with the corresponding delta and confidence threshold as the first run. This anonymized data provides a larger number of pseudonyms to be marked as traveled between nodes, which opposes the implementation in Yieh’s thesis [44]. Previous implementations have used the anonymized data, so we aim to confirm the claims made by Yieh that using de-anonymized data for training is more accurate [22].

Figure 26 and Figure 27 shows a slight improvement in both precision and recall in using de-anonymized data compared to raw identifiers. The one outstanding data point where the anonymized data has a slightly better precision is linked to the transitive property in the Z2Z matrix. A single association that linked two groups of Zigbee pseudonyms to each other was made at this step, leading to a higher precision. Since the majority of the differences come from the Exit-to-Enter attack application, we can look back at the weightings applied to the edges of the bipartite graph as an indicator to the variations.
Stepping in to the Exit-to-Enter attack application, the bipartite graph uses edge weights to decide on associations. These weights are determined by the behavior of the target device. In order to understand the behavior of the target device, we need to understand the context of the target device carriers. Every carrier of the device was a student at Cal Poly. This immediately affects the behavior of the target device, as the target device is now confined to the schedule that a student would follow while attending Cal Poly.

As an example, we use target device data from February 27th to explain erratic behavior. We first see the target device appear near the library and it shortly disappears. Six minutes later, we see the target device at the Pilling building for multiple hours (student attending a lecture). This is followed by the carrier “sprinting” across campus in less than two minutes to reach their next class caught by a probe in the Engineering IV building, where the device is again immobile for another period of time.

This example can be compared with the behavior of the other tracked devices to find that the time taken to travel between nodes highly depends on the day, time, carrier, and possibly even weather (it rained 6 of the tracking days). These all can explain the erratic behavior of travel between nodes. While it is possible to account for these factors using machine learning techniques, this was decidedly outside of the scope of this project.

For our application and dataset, we see multiple large gaps in time between the time when devices leave a listener a show up at the next one due to the target device carriers. This leads to minimal improvement in results, as both the anonymized and raw identifiers have erratic behavior for travel between nodes.
10.5 Path Reconstruction

Yieh proposed that an attacker could use the tracking data to recreate pathways taken by vehicles through a city. We apply the same analysis in this paper to show that even though this is possible, it is not always accurate.

In Figure 28 there is a comparison of the reconstructed path compared to the actual path taken for one of our target devices traversing campus as seen by the tracking network after associations have been made. The actual path was recorded by the device carrier for the day of March 5th. Times are not included in the figure to allow for clarity of location.

Discrepancies can be explained by missed pseudonym associations, and observation intervals that were classified as possible associations, but did not meet the confidence threshold in order to be determined as an actual association. While the reconstruction process is not perfect, we can see that there is obvious path that the target device takes though the campus. Even though this is only a single day of path reconstruction, an attacker can extrapolate this data to determine a target’s typical weekly schedule.
Figure 28: Reconstructed Path vs. Actual Path
We cover future work to supplement the implementation of this thesis as well as overarching discussion that comes as a result of this project. In the real world we see a major problem of missing information and loss of packets that are not seen in simulation. Following this, we cover further tuning of the tracking algorithm and improvements that could be made on the current implementation.

11.1 Dealing Missing Information

This section aims to cover discrepancies in the obtained data set. There are instances in the data set where given WiFi addresses can not be matched to their associated Zigbee addresses. The given reasons for this are twofold.

The first of these were the responses from WiFi devices that had received a beacon from the tracking device. Each target device would send a broadcast frame, and any access point within range would respond. Due to the previous mentioned hidden terminal instance as displayed in Figure 12, there are instances in the data set where we see responses to the target device WiFi heartbeat, but no packets sent by the target device.

The second of these was the possibility that a target device proceeded through the coverage area of a listener, but was not in range long enough to obtain a packet from both tracking vectors. There are instances in the data set where a device enters a coverage area, a single Zigbee packet is recorded, and then the device leaves. This
same situation applies to single WiFi packets seen. An instance where information is gained from a Zigbee broadcast, but not the WiFi broadcast is depicted in Figure 29.

Since pseudonyms are applied to Zigbee packets, a lone WiFi packet does not create an issue as we have a direct mapping to the device. While these single packets do not have any significant meaning by themselves, we see that they can be used to track a device’s path through the tracking network.

In the case of a lone Zigbee packet, this creates a significant problem in mapping the Zigbee pseudonym to a WiFi address. In some cases we can track a pseudonym in time to another listener that allows for a proper association to either another Zigbee pseudonym or directly to a WiFi address. If none of the attack strategies lead to the association of this Zigbee pseudonym, the pseudonym is marked as ”not enough information” to perform an association. Previous work has attempted associate these pseudonyms to no avail [44], and has been determined to be out of the scope of this thesis.
11.2 Variable Heartbeat Rates

In our implementation, we do not allow for the target devices to vary their heartbeat rates. We set a heartbeat rate of 5 seconds for the WiFi communication protocol, and 15 seconds for the Zigbee communication protocol. The attack model utilizes this information as a means to determine the lengths of observation periods and determine the plausibility of pseudonyms being associated.

There are possible variations that could be studied to better understand the strengths of the attack model. This first of which is to allows for the same tests to be run, only with a different delta between heartbeats for the communication protocols. By varying the heartbeat rates, we would be able to see if the attack model holds up regardless of the rate at which information is received by the listeners.

Secondly, we could test a heartbeat rate range. This would allow for a WiFi heartbeat to be sent every 3-7 seconds rather than the consistent 5 second intervals. While this is not an entire layer of security, a large enough range has the possibility of disrupting some aspects of the Exit-To-Enter attack and the Same Listener attack.

11.3 Increasing Number of Tracked Devices

This study utilizes four tracked devices over the span of 3 weeks. While this implementation was limited by resources, the simulation that this project was based on included over 200 tracked devices [44]. By increasing the number of tracked devices to the same scale as the simulation, we could possibly have a less successful pseudonym recall percentage based on the volume of pseudonyms generated by all of the new devices. In addition, we could see the effect that the number of devices tracked has on the attack model.
11.4 Adding Additional Tracking Vectors

This thesis only employs two tracking vectors, WiFi and Zigbee. The attack model can be generalized to an n-vector attack model. Ideally, this would provide more information on each target device. Consider a device that turns off WiFi and switches to the cellular network once the device if the user is in a WiFi dead zone. Once the device switches over, our listener implementation loses a significant portion of information. By expanding the listener implementation to a more refined cellular tracking interface, we could obtain more information on the location of the target device. Modifications to the attack model association matrices would have to be changed to support additional tracking vectors.
Preserving location privacy in the evolving technological landscape is already threatened by the attack model presented in this paper. In order to protect against these threats, we must first understand how our current wireless communications fail the end user. In this thesis, we document a real-world implementation of an attack model that has only been simulated in the past.

Using commercial off the shelf hardware, we implement a tracking network on the California Polytechnic State University campus that tracks target devices using WiFi and Zigbee communication protocols. Using a pseudonym-switching scheme applied to the collected data, we employ an attack model that allows for association across anonymized data in order to de-anonymize users and ultimately reconstruct their paths through the Cal Poly campus.

This thesis documents the feasibility of implementing a low-cost tracking network in order to track end users and threaten their location privacy. This case study is a building block for future research of the proposed attack model. By showing the strengths and weaknesses of the proposed attack model, more privacy preserving wireless communication protocols can take preventative measures to protect their end users.
BIBLIOGRAPHY


   https://www.home-assistant.io/components/deconz/.

   systems (srs). https://github.com/srsLTE/srsLTE.


   boards/development-boards/raspbee/description/?L=1.

[7] Zshark. https://www.dresden-  
   elektronik.de/funktechnik/products/software/zshark/?L=1.

[8] Google is probably tracking your location, even if you turn it off, says report -  
   cnet. https://www.cnet.com/news/google-is-probably-tracking-  
   your-location-even-if-you-turn-it-off-says-report/, August 2018.

   https://www.statista.com/topics/2711/us-smartphone-market/,  
   2018.

[11] 3GPP. evolved universal terrestrial radio access (e-utra); user equipment (ue) procedures in idle mode.


