DeTagTive: Linking MACs to Protect Against Malicious BLE Trackers

Tess Despres
tdespres@berkeley.edu
University of California, Berkeley

Noelle Davis
noelledavis@berkeley.edu
University of California, Berkeley

Prabal Dutta
prabal@berkeley.edu
University of California, Berkeley

David Wagner
daw@berkeley.edu
University of California, Berkeley

ABSTRACT

Bluetooth Low Energy (BLE) location trackers are popular and useful for finding misplaced keys, devices, and other items. However, they can also be used to track people and enable abuse. The companies that make location trackers, such as Apple, have worked to address these issues by adding notifications within their ecosystem and publishing recommendations for other manufacturers, creating a vertically integrated solution. More generally, however, BLE devices which rotate their MAC addresses remain elusive to responsible detection within and across many manufacturer platforms. Rotation of MAC addresses is crucial in some non-malicious usage scenarios for the privacy of the device owner, but this feature also makes detection more difficult. In this work, we propose and evaluate a detection algorithm that is robust to rotating MAC addresses by using parameters that appear to offer implicit continuity, including signal strength and advertisement intervals. A preliminary test of our algorithm, on four common BLE trackers and across multiple scenarios, shows this approach to be promising and practical.

CCS CONCEPTS

• **Security and privacy** → **Social aspects of security and privacy**; **Mobile and wireless security**; • **Human-centered computing** → *Ubiquitous and mobile computing systems and tools.*

KEYWORDS

privacy, bluetooth, tracking, wireless security

ACM Reference Format:
Tess Despres, Noelle Davis, Prabal Dutta, and David Wagner. 2023. DeTagTive: Linking MACs to Protect Against Malicious BLE Trackers. In *Second Workshop on Situating Network Infrastructure with People, Practices, and Beyond (SNIP2+ '23), September 10, 2023, New York, NY, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3609396.3610544

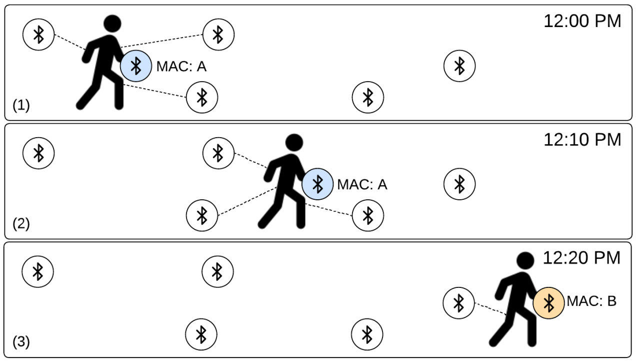

Figure 1: When traveling with a BLE location tracker, the device MAC and RSSI can be used to detect malicious trackers. Current approaches assume a static MAC address which would be successful from frame (1) to frame (2), but would not detect the device upon rotation in frame (3). Our approach aims to help with this limitation by providing a method to detect devices across MAC rotations.

1 INTRODUCTION

Pervasive use of Bluetooth Low Energy (BLE) devices is quickly becoming a day-to-day reality. This explosion in devices is in part driven by location trackers such as AirTags [2] and Tiles [5] which are popular BLE devices that help find missing objects. Location trackers result in a unique privacy and security challenge. They are designed to track objects, which means they can easily be misused to track people and enable abuse [16, 25]. Current approaches to mitigate this problem require device manufacturers to change the behavior of their tags, e.g., so they rotate MAC addresses infrequently or participate in special protocols to enable detection of misuse. However, this is challenging to deploy because it requires cooperation of all device manufacturers and does not address the population of existing devices.

In this paper, we demonstrate an approach that can be used to detect misuse of existing tracking devices, without requiring any changes to existing devices or cooperation from the device manufacturer. In particular, we demonstrate it is possible to link together rapidly rotating (e.g., every 15 mins) MAC addresses using transmission characteristics, such as signal strength and advertisement intervals, to detect a wide range of BLE location trackers. This algorithm, using no specialized hardware beyond what is found on a cell phone, allows detection methods to move beyond detecting trackers with semi-static MAC addresses.

Technology aided and enabled abuse is unfortunately neither new, nor uncommon [12, 19]. Smart home devices can be used as tools of domestic abuse with alarming ease. Cameras, locks, and

SNIP2+ '23, September 10, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0304-1/23/09.
https://doi.org/10.1145/3609396.3610544

thermostats can be controlled from apps remotely which means that devices can be used to remove access to spaces, watch, and intimidate others from afar [1, 25]. With mobile devices, this harmful surveillance can extend beyond the home. Location trackers, in particular, pose a threat because they are small in size and profile, making them easy to hide in clothing, cars, or bags [7, 16]. Unlike with mobile phones which are often password protected, it only takes a moment of physical access to plant a BLE location tracker in a bag or under a car in order to monitor someone from afar.

BLE tracking devices present a particular challenge because they serve a valuable purpose (e.g., finding lost items) but they must be designed to protect the privacy of both their owners as well as of third parties. For a non-malicious tracker carried by its owner, the uniquely identifying MAC address must be rotated not too infrequently in order to prevent monitoring of the owner's presence, for example detecting visits to a public location where there is a planted scanner. For a tracker that is surreptitiously and maliciously planted on a victim to track them without their consent, we would like some way for the victim to detect the presence of the tracker; this is easier if the MAC address is static or rarely changes, since then the victim's phone can detect the presence of an unfamiliar device that moves together with them.

Apple and others have focused significant energy on fixing this problem by detecting and notifying people within their ecosystem of device trackers. In fact, Apple and Google announced an industry-wide proposal to address risks of BLE location trackers [18]. The proposal has devices rotate MAC addresses only every 24 hrs, while away from the device owner, which allows people to detect when they are being tracked using the device MAC address as an identifier [18]. However, many modern BLE devices, location trackers included, periodically rotate their MAC addresses much more frequently to prevent identification. Typically MAC rotation is crucial for privacy, and considered a feature, because it helps reduce the risk of persistent tracking (e.g., monitoring an individuals movement throughout a mall based on observing their phone's MAC address at different locations in the mall over time). However, in the case of malicious trackers (Figure 1) a rotating MAC makes it hard to detect the presence of a hidden tracker.

We challenge the assumption that having a constant MAC is necessary to detect malicious tracking. In fact, static MACs cannot be depended upon because some BLE location trackers, such as the Samsung Galaxy Smart Tag, do rotate their MACs frequently. A motivated attacker could easily order and use trackers which do not follow the guidelines. Instead of counting on a static MAC, we take advantage of inherent properties of BLE which are continuous across rotations, to detect devices. There is a set formula followed by BLE advertisements, used by these devices. The spectrum for BLE extends from 2402 MHz to 2480 MHz and in order to communicate, devices hop across 40 different channels at set frequencies. Three of these channels are reserved for advertising (CH37, CH38, CH39) [4]. Devices also advertise at set intervals, which typically do not change with rotation. Advertisements are limited to 39 bytes and must include a header (2-bytes) and MAC address (6-bytes) [4]. Advertisements also can but do not necessarily include different data types including manufacturer data, TX power, local device name, service universally unique identifiers (UUIDs), and service data. For our algorithm, we use only the MAC address and received signal strength indicator (RSSI), which is supplied by all BLE radios, along with timestamps to fingerprint devices while being robust to manufacturer specific variations.

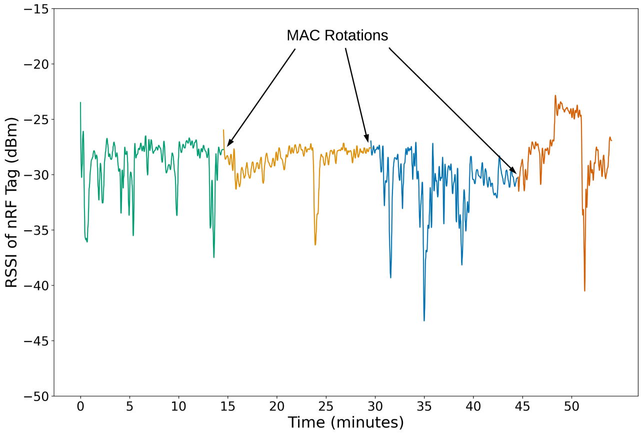

Figure 2: While traveling with our nRF device and BLE sniffer, we recorded the RSSI and MAC addresses. Because, the signal strength is relative to location we can use the characteristics of the trace to link MACs across rotations, identifying our nRF device as a tracker.

Our experimental setup uses a Raspberry Pi based BLE sniffer to capture advertising data. The advertisement data is processed using RSSI strength and a window based algorithm (described further in Section 3) to link together MAC addresses which have rotated (example rotation shown in Figure 2). We analyze our system on four different popular BLE location trackers (AirTag, Tile, Samsung Smart Tag, and Pebble) and for walking, biking and transit scenarios. We also isolate and characterize each device using a Faraday cage in order to understand how trackers advertise. Overall, this paper presents and evaluates a method to detect persistent devices across MAC rotations based on signal strength. We believe this technique could augment and address a key limitation of current efforts to prevent malicious tracking.

2 RELATED WORK

The social aspects of technology enabled abuse and locations tracking are well explored research areas [9, 11, 13, 21–24]. Freed et al. [12] conducted focus groups with survivors of intimate partner violence (IPV) and interviewed IPV professions. They found that many attacks, including location tracking, are facilitated by off the shelf devices. This type of surveillance extends to online communities as explored by Tseng et al. [25] meaning that attackers often share techniques such as particular devices which avoid detection. The social aspect of tracking points toward a need for new techniques to enable more robust detection.

There has also already been substantial work in detecting malicious tracking devices by both fingerprinting BLE transmissions and identifying BLE devices over time [8, 14, 15], but these approaches either depend on specialized RF sniffing hardware, pre-existing knowledge on how MAC addresses rotate, or parsing out specific fields in the advertised packet. Others simply do not track across ID rotations. Device fingerprinting based on signals is also a common method to detect 802.11 devices [10, 17]. Our proposed work would enable continued identification across BLE MAC rotations without any specialized hardware or information about other advertisement field values.

There are several responsible detection applications currently in use for malicious trackers. Apple has released an app for Android users, called TrackerDetect [3] with the limited scope of detecting malicious AirTags on Android devices. Another application, AirGuard [15], periodically scans for BLE devices, stores their broadcasted MAC addresses, and notices when a particular address is seen repeatedly, but their app also does not associate device identity across different broadcasted addresses. With MAC addresses used as fixed device identifiers, the algorithm has to detect a device from scratch after an address rotation. Finally, the BLE-Doubt [8] application constructs the trajectory of a BLE device over time using GPS data collected during sniffing and uses a combination of travel duration and distance to determine whether to mark a tracker as suspicious. Our algorithm is tangential to these efforts and could be added to existing solutions enabling detection of a wider range of BLE devices.

In a promising effort, Apple and Google released an announcement of their intention to work together on an industry specification [18] to combat the malicious use of location trackers. In the proposed specification, as is required of many continuously advertising BLE devices, a tracker traveling nearby its owner would be mandated to rotate MAC addresses on a 15 minute period to deter tracking of the owner. However, when the tracker is separated from its owner, this is lengthened to a 24 hour period in order to enable another smartphone to identify the device. Location trackers which are near their owner, according to recommendations, should also not advertise to reduce false positives. If all location trackers followed these design guidelines, it would certainly increase the ease of detecting malicious trackers, but the status quo is rotation across the broad BLE ecosystem, and it seems likely some trackers will continue to rotate.

Along with the MAC address, an advertising packet can also contain other fields specific to the manufacturer and device. One study [6] has shown that in particular, the manufacturer data field of the advertisement sometimes has components change asynchronously with the MAC ID rotations, allowing for correlation across different MACs. This method was introduced as an attack and therefore many devices have fixed this problem. Additionally, this method relies on manufacturer data fields which are subject to change over time [6]. In our case, we design an algorithm that takes advantage of the proximity of the device to break MAC rotation, which notably does not break the privacy properties of MAC rotations in cases where the device is not continually present.

On the other end of the spectrum, there has been work on identifying BLE devices without using the actual content of their advertisements, but rather by the "fingerprint" of their individual physical-layer hardware imperfections via measurements of RF features like carrier frequency offset (CFO) and I/Q offset [14, 20]. This approach, while applicable to diverse BLE devices with high short-term recognition accuracy, requires specialized RF sniffing hardware and suffers from loss in classification accuracy over the passage of time.

```
remove traces that are too weak (mean(RSSI) < theshold RSSI)
remove traces that are too short (length < min length)

for each trace which ends:

  window = 10 * average advertisement period
  candidate links = all traces which start in window
  filter out candidate links with long advertisement periods

  for each candidate link:
    compute match score (Eq. 1)

  best match score = min(match score)

  if the best match score is below a theshold score:
    link found!

  repeat looking for next match on the link
  until no more links on the full trace are found

  if net length with links > theshold length:
    flag as tracker!
```

Figure 3: Pseudocode of algorithm for classifying a device as a suspicious tracker. MAC addresses which match in characteristics are linked together, and if present for over a time length threshold, added to a list of suspicious potential trackers.

Significant work has been done in developing systems to identify BLE devices across MAC rotations using specialized RF sniffing hardware and data structures sleuthed from proprietary manufacturer-specific data fields. There are also existing smartphone-based scanners to increase awareness of BLE trackers one may be carrying unwittingly, which rely on static MAC identifiers. However, a need still exists for a reliable scanner without highly expensive hardware requirements that is not duped by MAC address rotations. To that end, we aim to design an algorithm that links together MAC rotations in a tracking case, which could be integrated into existing detection software.

$$\begin{aligned} matchScore = {} & |meanRSSI_{trace} - meanRSSI_{candidate}| \\ & + |stDevRSSI_{trace} - stDevRSSI_{candidate})| \\ & + |meanAdvPeriod_{trace} - meanAdvPeriod_{candidate}| \end{aligned} \tag{1}$$

3 ALGORITHM DESIGN

The algorithm links rotated MAC addresses that belong to a single device in order to determine if a device has been present for a significant amount of time. In order to link MAC addresses belonging to the same device, we take advantage of other characteristics of the trace which do not change, in particular the signal strength stays high as long as the device is nearby and the advertisement profile does not change. Therefore, when a trace disappears we look ahead for a potential match. More specifically, we compute a match score, shown in Equation (1), between the final ten advertisements of the first fragment and the initial ten advertisements of the candidate for the second fragment. We use the absolute value of the difference between mean RSSI strength, RSSI standard deviation, and mean

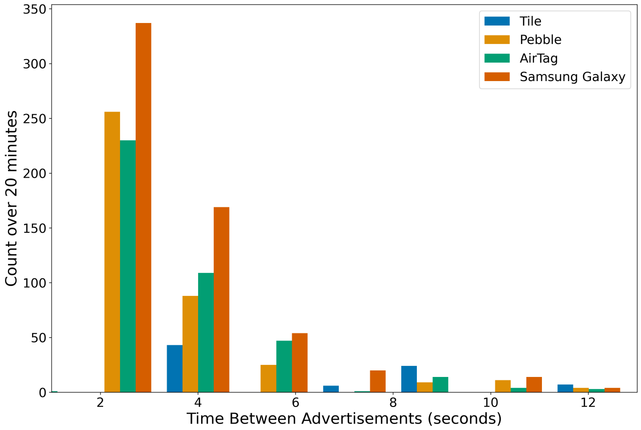

Figure 4: Example advertisement intervals for location trackers. This shows that for example the Samsung Galaxy advertises predominantly at 2 second intervals with some longer intervals likely due to missed advertisements. Airtags, Tile and Pebble have similar, but distinct patterns, with Tile having the longest advertisement period of around 4 seconds.

advertising interval in seconds. The match score is computed as the sum of these factors.

The fragment with the lowest match score among all link candidates, if above an experimentally determined threshold, is linked to the first fragment, and then the process repeats using the combined trace as a new fragment. This algorithm is laid out in pseudocode in Figure 3. We use a variable time threshold of 10 to 40 minutes with an average RSSI strength above -80 dB and minimum length of 8 minutes as hyperparameters to determine whether to mark a device as a suspicious tracker. For our threshold score we used a large number (1000) to encourage matches. This could be tuned down to reduce the risk of false positives.

To understand how different devices work, we isolated each device in a Faraday cage. We found that the devices advertise at set intervals. The device advertisement patterns are shown in Figure 4. The Galaxy Tag, for example, advertises predominantly at intervals of 2 seconds. Longer times between advertisements could be due to either missed packets or pauses in advertising. The other three tags we measured followed a similar advertisement pattern with some variation in the predominant interval (up to 5 seconds). This is interesting and useful because it allows us to filter out devices with much longer advertisement intervals (e.g AirPods). In terms of rotation time, the Samsung Galaxy Tag is the only tag that we observed rotations for (every 15 minutes). The other devices rotate on longer time scales, which is in line with the Apple/Google recommendations [18].

4 EXPERIMENTAL SETUP

The test system consists of a BLE sniffer, a simulated tag with operator-designed advertisements and MAC rotation interval, and several commercial tags (AirTag, Tile, Pebble, Samsung Galaxy Tag). The sniffer is implemented on a Raspberry Pi using the Noble module (Node.js BLE central module) to scan for and parse public, unencrypted BLE advertisement data into a log file. The logged data includes MAC address and received signal strength of the advertisement (RSSI), as well as a few other advertising data components which we use for sanity checking the links: connectability, public name of the device (if provided), transmitted power level (if provided), manufacturer data (if provided), and the first available service UUID (if provided). The Raspberry Pi is powered by a portable charger for mobile testing and various settings.

The simulated tag is implemented as a BLE peripheral on a Nordic nRF52 development board and provides a ground truth comparison for the detection algorithm. The simulated tag has an advertising interval of 1 second and rotates to a new MAC address every 15 minutes. The sequence of MAC addresses is known and can be compared with sniffer-detected advertisements for verification of the detection algorithm.

Lastly, we validate our sniffer and detection algorithm with four commercial location tracker tags from various manufacturers (AirTags, Tiles, Pebbles, and Galaxy Smart Tags). These are each paired with a smartphone to put them in operational mode and then kept out of range of the phone they have been paired with so that they behave as they would away from the owner. We aim to answer two key research questions in our evaluation. How well does our MAC linking algorithm perform? And how long approximately does it take to detect devices without false positives?

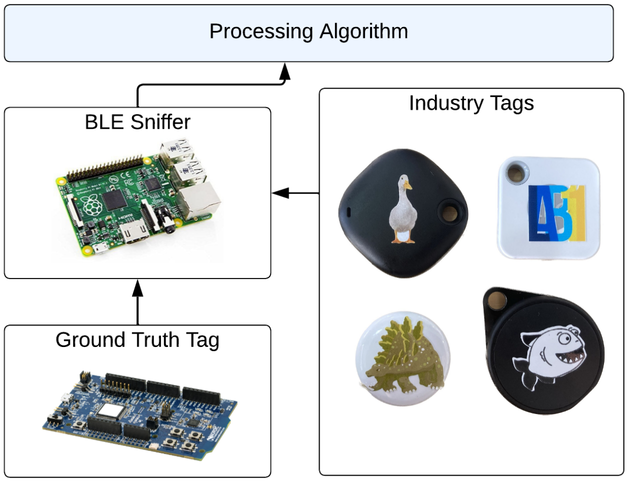

Figure 5: A high level overview of our system. We use a Raspberry Pi based JSON BLE sniffer to gather data from both industry tags and our own nRF52-DK based BLE Tag. Data from our sniffer is fed to a custom algorithm which detects potential location trackers.

5 RESULTS

In this section, we test our algorithm with our ground truth nRF device, evaluate our system with the four different industry BLE location trackers, and finally show how much time it takes to eliminate false positives and detect trackers while walking and biking.

Scenario	Minimum Time	St. Deviation Time
Walking	21 mins	0.4 mins
Biking	16 mins	2.32 mins

Table 1: We measured the minimum and standard deviation for the time length thresholds across our 5 tests in each scenario for identification of all three devices. This gives an estimate for how long is necessary to eliminate false positives in our three scenarios.

5.1 Algorithm Performance

We measure the algorithm in terms of how many links are successfully matched between trace segments. For example, if a device rotates every 15 minutes over an hour we expect the algorithm to stitch together four segments with three links for a perfect reconstruction of the RSSI trace. Because, we programmed the nRF device to rotate with consecutive MAC addresses, we are able to check how many nRF device links succeeded compared with the perfect ground truth advertised data. Our linked nRF trace missed a total of 2 links across all five 50 minute walking tests and 1 link across all five 45 minute biking tests giving a failure rate of 10.5% and 6.3% respectively. All three links that the algorithm missed were at the very end of the log file (i.e when the MAC rotated and the test abruptly ended shortly after rotation). This is not unexpected because if the trace is too short (under 8 minutes) the algorithm filters it out, and because the missing link is at the end of the sequence, it does not impact the number of devices detected. The nRF device is correctly detected as a present tracker across all tests.

5.2 Device Based Tests

With each BLE tag, we walked around campus for one hour with a location tracker in a pocket and our detection device in our backpack. This method allows us to simulate a realistic tracking scenario. We also carried our ground truth nRF52 tag in our backpack as a reference point. The large amount of BLE traffic from devices we encounter makes tracker detection challenging. Our sniffer picked up traffic from the participants own devices, our nRF52 tracker, the BLE location tracker of interest along with background traffic from surrounding devices.

We ran a 1 hour walking test with all four devices and were able to detect each device with the full hour of data and a threshold length set to 30 minutes to classify as a tracker. In our tests, the AirTag, Tile, and Pebble devices did not rotate their MAC addresses. The Galaxy Smart Tag rotated MAC addresses approximately every 15 minutes. Our algorithm also flagged the participant’s phone as a tracker, which rotated every 15 minutes as well.

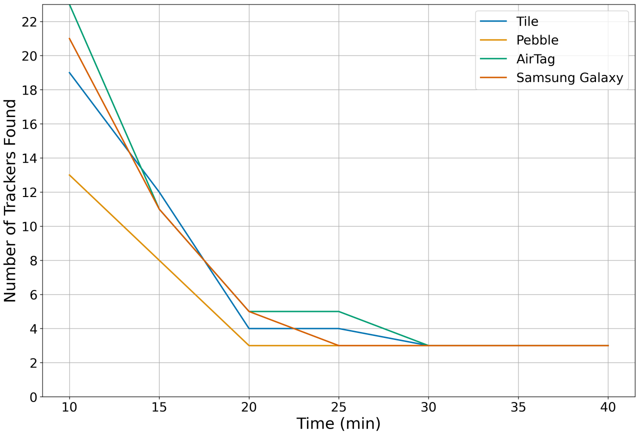

Figure 6: The number of devices detected by our algorithm based on the amount of time. Our algorithm accurately detects the three devices traveling with the participant with a 30 minute threshold for all four devices tested.

We tested our algorithm with different thresholds lengths to estimate how long it takes to detect devices that are following a participant (results in Figure 6). We expected that with longer time intervals there would be fewer false positives. When we modify the threshold length to be 10 minutes, we find that between 13 and 23 devices are present around the participant for 10 minutes or more over the entire test segment. This over estimate makes sense because devices in the surrounding area are flagged as potential trackers before we have time to move away from them. With longer time thresholds, fewer devices are flagged as traveling with us. All four devices in our test resulted in 3 trackers with a 30 minute threshold. The three detected devices are the nRF52 device, the participants phone, and the BLE location tracker.

5.3 Scenario Based Tests

In addition to testing with different devices while walking the same path on campus, we also tested our algorithm with variable thresholds while walking and biking. Finally we tested while taking public transit and observed false positives due to others traveling with us.

For our walking tests, we walked in 45 minute intervals across both crowded downtown areas and forested paths. We collected 5 walking tests in total with the Samsung Galaxy Tag in a pocket or bag and our detecting device in our bag. The Samsung Galaxy Tag was chosen as the device to test with because of the rapid MAC rotations. In addition to the Samsung Galaxy Tag, we also carried our nRF device and cell phone. In all five tests, the algorithm correctly identifies the three devices within 25 minutes, shown in Figure 7. Similarly in our biking tests, we followed different paths in both urban and rural areas for 45 minutes each, shown in Figure 8. The minimum time length threshold and the standard deviation threshold are shown in Table 1.

In our transit tests, we carried the Samsung Galaxy Tag in our pocket while riding AC transit for 45 minutes and while riding BART for 1 hour. In this scenario, we were able to detect the Samsung Galaxy Tag and link across rotations, but we also observed multiple false positives even with a threshold of 35 minutes. This brings up an important limitation of our approach, if we travel with other people, the algorithm will also detect and attempt to link together their devices including cell phones. We discuss this along with other challenges in Section 6.

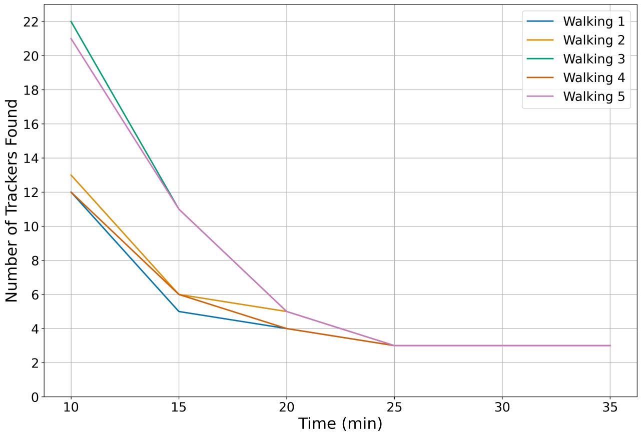

Figure 7: The number of devices detected by our algorithm based on the time length threshold while walking with the Samsung Galaxy Tag. Our algorithm accurately detects the three devices traveling with the participant with a 25 minute threshold for all five walking tests.

6 LOOKING FORWARD

Our algorithm and setup is not a complete solution, and is designed to supplement the existing work in the area. In particular, choosing when to notify and what is a tracking scenario remains challenging, and should be done with care to prevent harmful false alarms.

Real-Time. To be useful to people, giving timely notifications is crucial. Our current implementation of our algorithm works in post processing. However, it could be integrated into an existing application to check the logs and link MACs every couple hours. The linked MACs could then be fed into the existing applications and treated as the semi-static MAC addresses when determining if the device is a tracker.

Owned Devices. During tests, we also detected our phone as a "tracker". To avoid detecting other devices that are owned, people need a way to mark devices as safe and recognized. One way to do this could be to use the OS to query for previously paired BLE devices. We could also give people an opportunity to mark a particular device as being safe each time it reappeared. This is an important feature because frequent false positives could inundate a user with annoying notifications which mask real risk.

Advanced Adversaries. Our algorithm and existing detection applications [8, 15] are designed to detect off the shelf devices which are not engineered to track people. Because our approach relies on RSSI characteristics, an advanced adversary could still avoid detection by changing the frequency and transmit strength of the device at the same time as MAC rotation. Fortunately, the majority of attacks are "UI-bound" and not carried out by an attacker with access to custom devices and firmware [12]. However detecting in this case as well would be interesting to explore in the future, perhaps using hardware specific radio information such as clock skew.

Tracking is Social. To track another person maliciously is inherently social and not only about the underlying information. For example, the same information (location over time and space) could be freely shared with a trusted friend, or even implicitly shared with a stranger who happens to be taking the same path. Co-location of devices is a technical challenge distinct from when to notify a user about the possibility of tracking. Being robust to a range of different cases is a difficult and open challenge in the space of detecting trackers, Apple's detection mechanism for AirTags, for example, often gives false positives when traveling with other people. To help with this case, it could be useful to use side channel information including the phone GPS (for path analysis) as BLE-Doubt shows [8], the phone accelerometer (to avoid detecting when stationary), or user interaction to give more information about the scenario. Deciding how to notify could also benefit from user studies for behavior analysis, or social graph information such as understanding if a user is with a trusted friend. In all of these cases, privacy challenges caused by using additional side-channel information should also be considered.

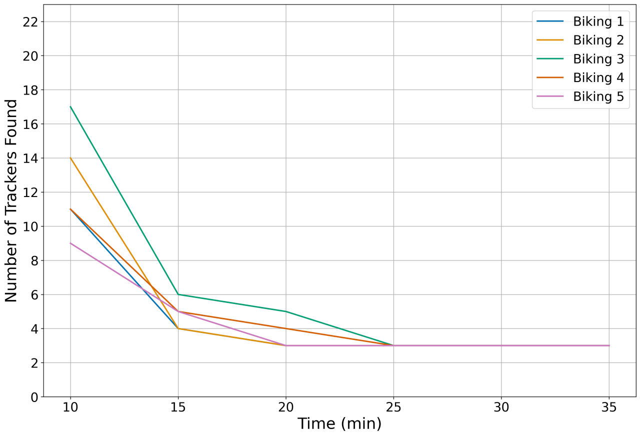

Figure 8: The number of devices detected by our algorithm based on the time length threshold while biking with the Samsung Galaxy Tag. Our algorithm accurately detects the three devices traveling with the participant with a 25 minute threshold for all five biking tests.

7 CONCLUSION

In conclusion, we present our algorithm to detect location tracking devices using MAC addresses and RSSI. We show that it works with four different common BLE tracking devices and in different scenarios albeit with false positives in the transit case. In the future, we hope this type of algorithm can be added to existing applications or software which detect malicious trackers. Acknowledging that the problem is not completely solved and impacted by social factors, we also give ideas for future work and discuss open challenges.

ACKNOWLEDGMENTS

Thank you to Alvin Tan for performing a literature review, Jean-Luc Watson for helping with the initial BLE sniffer design, and to the entirety of Lab11 for giving feedback.

REFERENCES

[1] 2018. Thermostats, Locks and Lights: Digital Tools of Domestic Abuse. https://www.nytimes.com/2018/06/23/technology/smart-home-devices-domestic-abuse.html.

[2] 2023. Airtags. https://www.apple.com/airtag/.

[3] 2023. Apple Tracker Direct. https://g.co/kgs/FhbU6y.

[4] 2023. Bluetooth technology overview. www.bluetooth.com.

[5] 2023. Tile. https://www.tile.com/.

[6] Johannes K Becker, David Li, and David Starobinski. 2019. Tracking Anonymized Bluetooth Devices. *Proceedings of Privacy Enhancing Technology* (2019).

[7] Alastair R Beresford. [n. d.]. Can’t Keep Them Away: The Failures of Anti-Stalking Protocols in Personal Item Tracking Devices. ([n. d.]).

[8] Jimmy Briggs and Christine Geeng. 2022. BLE-Doubt: Smartphone-Based Detection of Malicious Bluetooth Trackers. *2022 IEEE Security and Privacy Workshops (SPW)* (2022), 208–214.

[9] Rose Ceccio, Sophie Stephenson, Varun Chadha, Danny Yuxing Huang, and Rahul Chatterjee. 2023. Sneaky Spy Devices and Defective Detectors: The Ecosystem of Intimate Partner Surveillance with Covert Devices.

[10] Jason Franklin and Damon McCoy. 2006. Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting. In *USENIX Security Symposium*.

[11] Diane Freed, Sam Havron, Emily Tseng, Andrea Gallardo, Rahul Chatterjee, Thomas Ristenpart, and Nicola Dell. 2019. "Is my phone hacked?" Analyzing Clinical Computer Security Interventions with Survivors of Intimate Partner Violence. *Proceedings of the ACM on Human-Computer Interaction* 3 (2019), 1 – 24.

[12] Diane Freed, Jackeline Palmer, Diana Elizabeth Minchala, Karen E. C. Levy, Thomas Ristenpart, and Nicola Dell. 2018. “A Stalker’s Paradise”: How Intimate Partner Abusers Exploit Technology. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018).

[13] Andrea Gallardo, Hanseul Kim, Tianying Li, Lujo Bauer, and Lorrie Faith Cranor. 2022. Detecting iPhone Security Compromise in Simulated Stalking Scenarios: Strategies and Obstacles. In *SOUPS @ USENIX Security Symposium*.

[14] Hadi Givehchian, Nishant Bhaskar, Eliana Rodriguez Herrera, Hector Rodrigo Lopez Soto, Christian Dameff, Dinesh Bharadia, and Aaron Schulman. 2022. Evaluating Physical-Layer BLE Location Tracking Attacks on Mobile Devices. *2022 IEEE Symposium on Security and Privacy (SP)* (2022), 1690–1704.

[15] A. Heinrich, Niklas Bittner, and Matthias Hollick. 2022. AirGuard - Protecting Android Users from Stalking Attacks by Apple Find My Devices. *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (2022).

[16] K. Hill and P. T. Heisler. 2022. I used Apple AirTags, Tiles and a GPS tracker to watch my husband’s every move. (2022).

[17] Tadayoshi Kohno, Andre Broido, and Kimberly C. Claffy. 2005. Remote physical device fingerprinting. *2005 IEEE Symposium on Security and Privacy (S&P’05)* (2005), 211–225.

[18] Brent Ledvina, Zachary Eddinger, Ben Detwiler, and Siddika Parlak Polatkan. 2023. *Detecting Unwanted Location Trackers*. Internet-Draft draft-detecting-unwanted-location-trackers-00. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-detecting-unwanted-location-trackers/00/ Work in Progress.

[19] Tara Matthews, Kathleen O’Leary, Anna Turner, Manya Sleeper, Jill Palzkill Woelfer, Martin Shelton, Cori Manthorne, Elizabeth F. Churchill, and Sunny Consolvo. 2017. Stories from Survivors: Privacy & Security Practices when Coping with Intimate Partner Abuse. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017).

[20] Daniel Nilsson. 2022. *Identifying Bluetooth Low Energy Devices via Physical-Layer Hardware Impairments*. Ph. D. Dissertation.

[21] Michaela M. Rogers, Colleen M. Fisher, Parveen Azam Ali, Peter Allmark, and Lisa Aronson Fontes. 2022. Technology-Facilitated Abuse in Intimate Relationships: A Scoping Review. *Trauma, violence & abuse* (2022), 15248380221090218.

[22] Julia Slupska and Angelika Strohmayer. 2022. Networks of Care: Tech Abuse Advocates’ Digital Security Practices. In *USENIX Security Symposium*.

[23] Julia Slupska and Leonie Maria Tanczer. 2021. Threat Modeling Intimate Partner Violence: Tech Abuse as a Cybersecurity Challenge in the Internet of Things.

[24] Sophie Stephenson, Majed Almansoori, Pardis Emami-Naeini, Danny Yuxing Huang, and Rahul Chatterjee. 2023. Abuse Vectors: A Framework for Conceptualizing IoT-Enabled Interpersonal Abuse.

[25] Emily Tseng, Rosanna Bellini, Nora Mcdonald, Matan Danos, Rachel Greenstadt, Damon McCoy, Nicola Dell, and Thomas Ristenpart. 2020. The Tools and Tactics Used in Intimate Partner Surveillance: An Analysis of Online Infidelity Forums. In *USENIX Security Symposium*.

Received 12 June 2023; accepted 10 July 2023