BLE-Doubt: Smartphone-Based Detection of Malicious Bluetooth Trackers

Jimmy Briggs
Unaffiliated
jimmy@jimmybriggs.net

Christine Geeng
University of Washington
cgeeng@cs.washington.edu

***Abstract*—Stalkers can hide Bluetooth Low-Energy (BLE) trackers, like the Apple AirTag and Tile Finder, in their targets' clothing or vehicles to surveil their locations. Existing countermeasures to detect BLE-based stalking are promising but have several shortcomings: they only work against Apple products, they are slow to detect trackers, and there is no publicly available characterization of how well they work. We present an open-source, general method for detecting maliciously deployed BLE trackers. Our algorithm detects malicious devices in just a few minutes, whereas previous algorithms take hours or days. We show in a small but novel validation study that our algorithm performs with high precision and recall for most extant trackers, although AirTags pose additional challenges. Along with our algorithm and validation, we provide an open-source Android application capable of real-time detection of these devices. We also characterize the behavior of the AirTag and discuss the risk factors which make it particularly hard to detect. We conclude with a discussion for future work to make tracking devices safer for the public.**

***Index Terms*—Bluetooth, Stalking, AirTag, Tile, privacy, surveillance**

I. Introduction

Crowd-tracked Bluetooth Low-Energy (BLE) beacons such as Apple Airtags and Tile Finders were originally designed to find lost or stolen objects. The user attaches the beacon to an object and uses their phone to identify the beacon's location. Now stalkers hide these beacons in their targets' possessions to track their location [5], [6], [14], giving stalkers the opportunity to harass and control people. While domestic abusers have previously used electronics like smart phones and GPS transponders to commit intimate partner violence [7], [8], [16], crowd-sourced BLE beacons raise additional concerns because they are small, cheap, easy-to-use, and can last for years without a battery replacement [10], [24]. An attacker only requires brief access to their target's clothing, possessions, or vehicle to hide a BLE tracker and establish precise, real-time tracking for an extended duration. Figure 1 illustrates this attack.

While Apple has developed a countermeasure to alert iPhone users automatically if an AirTag is tracking them, Android users have to manually open Apple's Tracking Alert app to scan for devices [26]. These countermeasures also do not apply to Bluetooth beacons from other manufacturers. Furthermore, Apple has declined to tell the public how their algorithms work or how effective they are, and informal experiments by journalists have shown that it may take days for Apple's countermeasures to work at all [6].

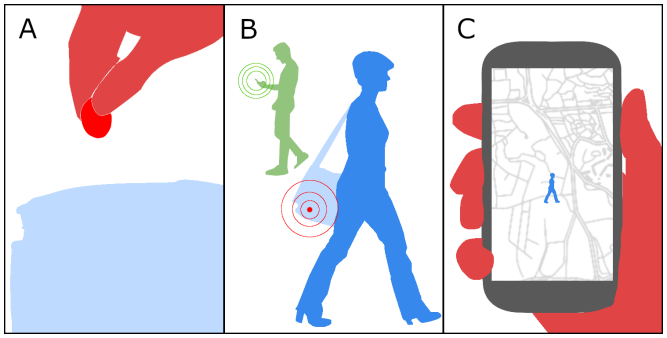

Fig. 1. The attack detected by BLE-Doubt. (A) The attacker places a tracker on the target's person. (B) The tracker is detected by a bystander's smart device and reported to the manufacturer's cloud. (C) The attacker monitors the position of the target with the manufacturer's smart phone app.

To rapidly and automatically detect BLE trackers, we developed BLE-Doubt, an open-source Android application[1] which scans for malicious BLE trackers using a novel algorithm based on a simple, topological model of tracking. In this paper, we present and validate our algorithm on a small new dataset generated by the authors through simulated stalking scenarios using a variety of real BLE trackers including AirTags, Tiles, and Chipolos. We show that our algorithm performs well with all trackers but the AirTag, which is particularly resilient to our countermeasures. We discuss the features of the AirTag which make detection particularly hard, and make design recommendations toward safer BLE trackers. We conclude with recommendations for future work.

II. Background and Related Work

A. BLE Trackers

Bluetooth Low Energy (BLE) is a low-power protocol for short-range radio communication between electronic devices in the 2.4 GHz ISM band [28]. A wide range of IoT devices communicate over BLE [3].

BLE supports several modes of operation, including the BLE "beacon." BLE beacons broadcast short messages called advertisements to all nearby BLE devices. These advertisements have been used for indoor localization [19], pairing

[1]BLE-Doubt source code is available at https://github.com/jeb482/bledoubt.

peripherals [28], and sharing point of interest information [21]. Beacon advertisements all include a Bluetooth MAC Address to identify themselves to listeners. In most cases, this address is static, but some beacons employ "Privacy Mode," in which the MAC address periodically changes [28].

A subset of Bluetooth Low-Energy beacons are BLE trackers, which use their advertisements to signal their locations. Typically, BLE trackers are short-range, but some manufacturers such as Tile and Apple have implemented a crowd-sourcing approach to localize them. In this approach, nearby smart devices may overhear the tracker's beacon advertisements. These smart devices then determine their own location (e.g. with a GPS) and inform the manufacturer's cloud that the tracker has been detected at that location.

Crowd-sourced localization of BLE trackers provides global, real-time tracking without decreasing battery life. However, it also allows stalkers and abusers to track their targets from any distance as long as the target occasionally passes by a device running the manufacturer's app. For example, every BLE-enabled iPhone may report AirTag locations to Apple's cloud.

B. Technology and Interpersonal Surveillance

Researchers have documented how abusers use technology to commit intimate partner violence (IPV) [8], [13], [16], [23], including through GPS trackers [25] or smart home devices [2]. Privacy experts have raised similar IPV concerns and concern over corporate abuse of privacy when Tile was released [17]. The National Network to End Domestic Violence also identified that Apple AirTags may put targets of IPV at extreme risk.

C. Existing Countermeasures

Apple's original "anti-stalking" solution for BLE trackers [11], released in April 2021, allowed iPhones to detect AirTags thought to be following iPhone users, but only when the AirTags were separated from their owners for some time. However, journalists found that Apple's countermeasures were insufficient to interrupt a simulated stalking scenario [6]. The specifics of Apple's algorithm are unknown.

In August 2021, the Secure Mobile Networking Lab published AirGuard, an open-source Android application designed to detect malicious AirTags [1]. AirGuard alerts the user when it detects a single AirTag in three separate locations. One potential weakness of AirGuard is that it may produce false positives if the user encounters a passerby with an AirTag in three different locations. In contrast to Apple's countermeasures, AirGuard's algorithm for detecting trackers is publicly known and testable.

In December 2021, Apple released the Tracker Detect app for Android. Unlike AirGuard, Apple's Tracker Detect requires the user to scan manually for AirTags, creating an additional barrier to detection [12]. Tracker Detect can force nearby AirTags to ring, making them easier to retrieve if they are detected. However, Tracker Detect's manual scanning requirement may decrease the chance that the trackers are detected at all. Neither countermeasure is effective against non-AirTag trackers.

Below we present BLE-Doubt, which automatically scans for and reports malicious trackers, is resilient to false positives, can detect trackers besides AirTags, and is easily extensible to new types of BLE trackers.

III. BLE-Doubt: A Tracker-Sniffing Application

BLE-Doubt is an open-source Android application that automatically identifies malicious trackers and alerts the user to their presence. At a high level, the BLE-Doubt app detects and parses BLE beacon advertisements, stores a history of these detections, classifies devices in the history, and alerts the user when a suspicious device is located. We refer to the history of a single beacon's advertisements along with the time and location of their detections as the beacon's *trajectory*. Much of BLE-Doubt's novelty comes from its trajectory classification algorithm, which determines whether or not a BLE beacon is suspicious based on its history.

A. Trajectory Classification

To classify a BLE beacon, BLE-Doubt must determine whether or not the trajectory of the device is "suspicious," i.e. following the user. In this section, we motivate and present our Topological Classifier algorithm along with three baselines for comparison and a trajectory decomposition. We evaluate our algorithm on the BLE-Doubt dataset in Section IV.

We model a beacon's trajectory R as a sequence of ordered pairs $\left(x_i \in \mathcal{S}^2, t_i \in \mathbb{R}\right)$ with the constraint that $\{t_i\}$ increases monotonically. $\mathcal{S}^2$, the unit sphere, represents the position of the device on Earth. $\mathbb{R}$, the Reals, represents the moment of a beacon advertisement. For brevity, we say that x_i is in R and t_i is in R if $(x_i, t_i) \in R$. The *diameter* of R is the largest great-circle distance $\rho(x_i, x_j)$ for any two positions x_i, x_j in R. Similarly the *duration* of R is the longest time $|t_i - t_j|$ between any two timestamps t_i, t_j in R.

Our baseline classifiers follow from these point-set statistics. Our first baseline, the Duration Classifier, classifies a device as suspicious if its trajectory has a duration which exceeds some threshold α. This evaluation is trivial in constant time because $\{t_i\}$ is monotonic. Similarly, the baseline Diameter Classifier classifies a device as suspicious if its trajectory has a diameter which exceeds some threshold β. We use the obvious $\mathcal{O}(n^2)$ algorithm to find the diameter, but more efficient methods are known [15]. We note the similarity of our Diameter Classifier to the AirGuard app [1]. Our final baseline, the Hybrid Classifier, labels a device as suspicious only when both the Duration and Diameter Classifiers would both classify it accordingly. We have found that 10 minutes and 300 meters work fairly well for α and β, respectively. We leave the tuning of these parameters to future work.

Our experiments show that all three baselines successfully detect malicious trackers, but produce too many false positives. Notably, no baseline can differentiate a device in constant proximity to the user from a device which the user happens

upon in multiple locations. By contrast, our Topological Classifier differentiates between these scenarios using the concept of ϵ-connectedness common in computational topology. ϵ-connectedness proceeds from an early formulation of connectedness by Cantor [9]. Our treatment of ϵ-connectedness is adapted from a 2000 paper by Robins & Meiss [20].

A trajectory is *ϵ-connected in time*—or ϵ-connected, for short—if no two sequential timestamps t_i, t_{i+1} are more than ϵ seconds apart. That is to say, there are no time gaps bigger than ϵ between consecutive beacon detections. Any device trajectory can be uniquely partitioned into a disjoint union of ϵ-connected subtrajectories [20]. Algorithm 1 provides such a decomposition in linear time. These subtrajectories are called ϵ-components. Each ϵ-component can be interpreted as a period of time during which the beacon was in continuous proximity to the user's device.

Algorithm 1: Partition a trajectory into ϵ-components

Input: A finite, non-empty sequence of monotonically increasing timestamps $\{t_i\}$ and a scalar duration ϵ.
Output: The collection $\mathcal{C}$ of index sets of the non-empty ϵ-components.

$\mathcal{C} \leftarrow \{\}$;
$\mathcal{I} \leftarrow \{1\}$;
for $j \leftarrow 2$ *to* $|\{t_i\}|$ **do**
  if $(t_j - t_{j-1}) < \epsilon$ **then**
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$;
  else
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathcal{I}\}$;
    $\mathcal{I} \leftarrow \{j\}$;
  end
end
$\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathcal{I}\}$;
return $\mathcal{C}$;

Our Topoplogical Classifier, provided in Algorithm 2, applies the trajectory statistics approach of our Hybrid Classifier to the ϵ-components of the device trajectory rather than the trajectory itself. We choose ϵ to be 3 minutes for all devices except the AirTag, which demonstrates long periods of inactivity. For the AirTag, we choose a much more conservative ϵ of 10 minutes. Future work should define a unique ϵ for each device model to match the empirical advertisement patterns of real devices.

B. Beacon Parsing

Before it can classify BLE beacon trajectories, BLE-Doubt must parse and record BLE beacon advertisements. BLE-Doubt parses beacon advertisements using the open-source Android Beacon Library [18]. Our beacon parser assigns every beacon a persistent identifier to keep track of its detections. We use the BLE beacon's Bluetooth MAC address—broadcast with each advertisement—as a persistent identifier. The MAC address is a good identifier for most devices, except for those in Privacy Mode, which periodically change or "rotate" their MAC addresses [27]. Of the devices we tested, only AirTags employed Privacy Mode, making them harder but not impossible to detect. We discuss this further in Section IV.

Algorithm 2: Topological Classifier

Input: A finite, nonempty device trajectory R, a constant closeness parameter ϵ, and two constant thresholds for duration (α) and diameter (β).
Output: A boolean value, true if and only if the device corresponding to R is identified as a suspicious tracker.

$\mathcal{C} \leftarrow$ Algorithm1(R.`timestamps`, ϵ);
for $\mathcal{I} \in \mathcal{C}$ **do**
  if *duration*$(R_\mathcal{I}) > \alpha$ && $diameter\,(R_\mathcal{I}) > \beta$ **then**
    return *true*
  else
    continue;
  end
end
return *false*;

Our parser uses the beacon's service identifier to differentiate potential trackers from other devices. This was determined empirically; the trackers we tested used recognizable service identifiers, presumably to declare themselves to their manufacturers' devices and apps. Devices with these identifiers are treated as potential threats, but not labeled suspicious until our Topological Classifier runs.

To detect Tile, Chipolo, Spot and AirTag trackers, whose header formats are not in the Android Beacon Library or publicly available, we used nRF-Connect [22], a BLE Scanner, to reverse-engineer their beacon formats. Currently, BLE-Doubt can detect the iBeacon, Altbeacon, Eddystone, Tile, Chipolo, Spot, and AirTag beacon layouts. Any beacon layout conforming to the BLE standard can be added by extending the library's `BeaconParser` class.

In contrast to the other devices we tested, Apple AirTag trackers encode their beacon advertisements as "manufacturer specific data" which is similar to other BLE-enabled Apple devices (e.g., iPhones and AirPods). This means BLE-Doubt must monitor all nearby Apple products broadcasting over Bluetooth in order to protect the user from maliciously deployed AirTags.

C. Database

The data storage for BLE-Doubt is segregated into two tables. The first table contains metadata about each detected BLE beacon which may represent a tracker. The second table contains spatiotemporal data of each detected beacon's trajectory.

The device trajectory is stored as a sequence of received signal strength indicators (RSSI) and latitude-longitude pairs indexed by timestamp and Bluetooth MAC address. This information encodes the time, location, and intensity of each

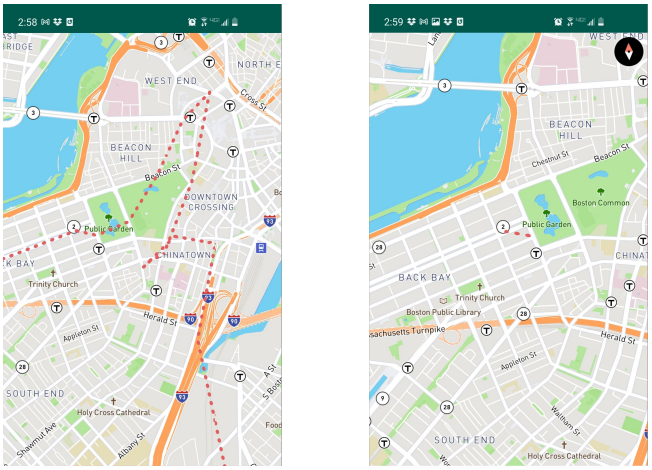

Fig. 2. A map of the trajectory of a malicious tracker (left) versus a false positive (right). The trajectories are displayed as dotted red lines. The user can compare these maps with their own memory of travel to determine whether a suspicious device was truly following them.

advertisement broadcast by the BLE beacon, and relates it to the appropriate metadata.

D. User Consultation

When a suspicious device is identified by BLE-Doubt's Topological Classifier, the user is alerted with a push notification. This notification directs to user to a visualization of the suspicious device's trajectory. This UI allows users to make a final decision of whether or not a suspicious beacon is malignant. A user can choose to mark the device as safe, which will prevent BLE-Doubt from sending more notifications. Alternatively, the user can elect to receive another notification if the beacon continues to follow them. If the device is still nearby, the user can use BLE-Doubt to scan for the device. An example of the map interface is shown in Figure 2.

IV. System Evaluation

Fundamentally, BLE-Doubt should classify BLE devices which follow the user around as "suspicious", and all other devices as "not suspicious".

It is vital that our algorithm exhibits a *high recall* to effectively identify dangerous trackers. Additionally, because the user is a non-expert with limited bandwidth, our algorithm must not overwhelm them with a large number of potential threats which turn out to be benign. Therefore our algorithm must also demonstrate a *high precision*. In this section we evaluate our algorithm's performance on a novel dataset, focusing on these two statistics. For each log, an optimal classifier should classify all the planted trackers as suspicious, and all other devices as benign.

A. Dataset

We logged sessions of BLE-Doubt usage in a major U.S. city corresponding to common modes of locomotion including walking, driving, and public transit. The logs are enumerated in Table I. Logs A-J were collected over a two-week period in May 2021. Each includes four to six trackers (drawn from a collection of paired Chipolo, Tile, and Spot devices, a programmable RadBeacon BLE Beacon, and unpaired AirTags) planted on the author. These logs constitute the evaluation set from which our classifier statistics were collected.

TABLE I
Logs in the BLE-Doubt Validation Dataset.

Log ID	Duration (H:MM)	Movement	Tracker Locations
A	1:15	Walking	Backpack
B	1:35	Walking	Backpack
C	1:15	Walking	Pockets
D	0:14	Walking	Pockets
E	1:24	Car	Vehicle
F	0:25	Jogging	Backpack
G	0:21	Walking	Backpack
H	0:35	Walking	Backpack
I	0:14	Train	Backpack
J	0:28	Train	Backpack

The researcher planted their own trackers in their clothing, possessions, or a vehicle, mimicking where an adversary could slip a tracker onto a target without their knowledge. Logs were collected by one researcher on their own phone. No additional participants were involved in the study. The authors intend to publish the BLE-Doubt dataset with this workshop paper, having scrubbed MAC addresses and device metadata from the dataset.

B. Classification of Suspicious Devices

We compared the performance of our Topological Classifier to that of three baselines on the BLE-Doubt dataset: the Duration, Diameter, and Hybrid Classifiers. For consistency, we chose the thresholds α and β in our topological classifier to match their counterparts in the baselines. Because our dataset was small, we chose not to tune our parameters empirically.

We found that each baseline had perfect or nearly perfect recall. The Duration Classifier's recall was particularly strong, identifying every malicious tracker in the dataset. On the other hand, each baseline produced a large number of false positives, as the confusion matrices in Figure 3 show. Of the baselines, the Duration Classifier was most likely to misclassify a device as suspicious, followed by the Diameter Classifier. While the true negative rate was quite high for each baseline, the disproportionate number of benign devices in the dataset meant that the small percentage of benign devices classified as suspicious exceeded the total number of malicious trackers of any label. This results in a precision below 0.45 for each.

Compared to the baselines, our Topological Classifier demonstrated vastly superior precision (0.94), but slightly inferior recall (0.92 versus 0.98, 1.00, and 0.98). Additionally, the Topological Classifier's F_1 score [4] is 0.93, compared with the 0.61 for the next-best classifier. The accuracy statistics for each classifier are provided in Table II.

Even with its improved precision, the BLE-Doubt Topological Classifier may occasionally misidentify benign trackers as suspicious. Therefore, we present the user with the final decision of whether or not a suspicious BLE beacon is in

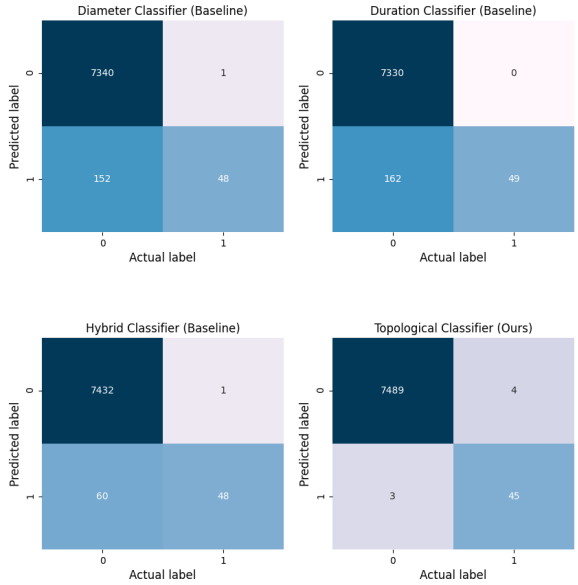

Fig. 3. Confusion matrices for baseline classifiers based on spatial diameter, duration, a combination of the two, and finally our topological classifier. The label "1" represents a suspicious device.

TABLE II
ACCURACY METRICS FOR EACH TRAJECTORY CLASSIFIER.

Classifier	Precision	Recall	F_1 Score
Diameter	0.24	0.98	0.39
Duration	0.23	1.0	0.38
Hybrid	0.44	0.98	0.61
Topological	0.94	0.92	0.93

fact malicious. We believe that a user who is concerned about the privacy of their location data may be willing to tolerate an occasional false positive, which they can validate based on their contextual information or by investigating their belongings for trackers. Future work should examine the trade-off between false positives and false negatives that users are willing to tolerate.

C. Operational Hardware Constraints

Even in an urban setting, the memory footprint represented by our database of devices remained small. The average JSON log in the BLE-Doubt format is 741 kB (standard deviation 685 kB), and we expect the SQLite implementation used on the smart phone to be even smaller. In terms of memory usage, in JSON format our logs occupied an average of 13.9 kB of memory per minute of logging (standard deviation 2.86 kB. Our current implementation of BLE-Doubt allows for manual database purges rather than automatic ones. It is a noteworthy side-effect of our classifier's design that a non-suspicious device can have its history cleared after any ϵ-component without affecting future classification fidelity. Given the large quantity of memory and disk space available to commodity smart phones, memory concerns should not affect the usage of BLE-Doubt.

Additionally, the battery consumption on the Samsung Galaxy S10 used to collect these logs was about 10% per hour running BLE-Doubt along with other essential tasks and an audio listening app. While this amounts to non-trivial battery consumption, it is not so extreme as to prevent a user from running BLE-Doubt altogether.

TABLE III
LOGS OF AIRTAGS DETECTED AMONG OTHER APPLE DEVICES.

Log ID	Duration	Movement	Apple Devices	Precision	Recall
K	1:30	Walking	iPhone, Watch	0.07	1.0
L	0:45	Walking	None	1.0	1.0
M	0:50	Train	Unknown, Many	0.015	1.0
N	1:04	Car	iPhone	0.25	1.0

D. Finding AirTags among Apple Devices

In addition to the BLE-Doubt validation set, we collected logs to study paired AirTags in the presence of other Apple ecosystem devices, which share identical beacon advertisement formats. These logs K-N, collected in January 2022, are described in Table III along with the precision and recall of the Topological Classifier ($\epsilon = 600s$) for each log. Each log represents a journey with two paired AirTags, collected in the same method as the BLE-Doubt validation set. Some of these logs also contain one or more non-AirTag Apple devices which happen to travel alongside the user. The Topological classifier exhibits flawless performance when no non-AirTag Apple devices travel with the user. However, when the user is accompanied by their own Apple device, or that of a another co-located traveller, precision drops proportionally to the number of co-travelling Apple Devices. Future work should explore ways to differentiate between benign, co-located Apple devices and maliciously deployed ones.

V. DISCUSSION

A. Efficacy of BLE-Doubt

Our analysis showed that BLE-Doubt's Topological Classifier detects threats about as well as the baselines, but with a vastly decreased prevalence of false positives, and an unmatched F_1 statistic of 0.93.

Given the large number of benign BLE devices in the modern world, we assert that a beacon tracker needs a low false-positive rate. Thus we think the improvement to precision and F_1 justifies the marginal loss of recall compared to the baselines. We hope future work will extend our methods by tailoring the algorithm to specific device formats.

We also note that like previous work, our algorithm requires that the user have relative freedom of movement and a smart phone. Thus users especially vulnerable to interpersonal surveillance, such as children, elderly people, people living in poverty, people experiencing domestic or family violence, incarcerated people, people living in group homes or hospitals, and people with mobility disabled by society may not be able to benefit from any mobile BLE tracker detection algorithm.

Therefore, tracking devices also need direct safety improvements and regulatory changes, which we explore below.

B. Toward Safer Trackers

While AirTags and other BLE Trackers have many benign uses, their use in stalking still presents a high risk. Here we make suggestions for manufacturers on how to make BLE trackers safer and easier to detect.

- *Use privacy mode, but rotate the Bluetooth MAC address infrequently.* Rapidly rotating Bluetooth MAC addresses limit the data available for detection algorithms like ours. We recommend a period of at least an hour, and preferably closer to a half day.
- *Announce Bluetooth MAC address rotations in a standard way.* Alternatively to the above, we suggest that the Bluetooth Standard be extended to allow devices to publicly announce what they are about to change their MAC addresses to. This would maintain a degree of privacy, as devices would become indistinguishable from each other when separated from an eavesdropper for more than one rotation, but would allow detectors like BLE-Doubt to correlate rotating MAC addresses in their constant presence over time.
- *Remove crowd-sourcing from BLE Tracker technology.* While this might reduce the user's capacity to find their lost or stolen keys, it would guarantee that trackers could not be used to stalk targets far from an attacker's own devices. BLE Tracker apps could still report when a device was last seen, but would no longer be able to track it after it first went missing.
- *Ensure that BLE-Trackers advertise frequently and consistently over time across all modes of operation.* This maximizes the ability of applications like BLE-Doubt to correlate data about the tracker over time, and may allow for less frequent scanning.
- *Focus on interventions that do not require targets to own or operate technology.* Trackers can be made large, loud, and three-dimensional to make it harder for an attacker hide them. Trackers can make sound moments after being separated from a phone rather than days.

Some of our suggestions may limit the utility of BLE Trackers, but this is by design. Maximizing the ability to locate lost or stolen property also maximizes the ability to surveil other people. Device manufacturers and regulators have the opportunity to decide what trade-off of these priorities best aligns with their values. Additionally, some suggestions may make these trackers less convenient and desirable. We believe that the inconvenience posed to a benevolent user by such changes is less significant than the potential harm of an undetectable tracker in the hands of an abuser or stalker.

C. Troublesome AirTag Advertisements

Compared to the other trackers we tested, AirTags were particularly difficult to detect. Tile, Chipolo, and Spot devices did not employ Privacy Mode, and advertised their presence every few seconds. Unpaired AirTags behaved much the same, but upon pairing, AirTags changed their behavior. While AirTags employ BLE Privacy Mode, this behavior did not pose an issue for our algorithm. AirTags rotate their MAC addresses infrequently—somewhere between every two hours and once a day. The Topological Classifier identifies suspicious devices quickly enough to avoid this issue.

More difficult was the erratic timing of paired AirTag advertisements. Whereas the other tested devices maintained a steady advertisement tempo after pairing, the AirTag would unpredictably go quiet for durations up to an hour during our tests. Moreover, when the AirTag resumed its advertisements, it would only do so for a few minutes at a time and would mostly remain quiet. This behavior persisted after we separated the AirTag from its host device. If this behavior is representative of AirTags in general, even the best possible BLE detection algorithm may be too slow to prevent harmful stalking—an hour can be long enough for harm to occur.

VI. Limitations

While the Topological Classifier itself is highly effective, we acknowledge that the operational constraints of smart phones complicate our success. Our system requires frequent if not constant BLE scanning to obtain the density of data required for the classifier. This imposes a heavy cost on the battery life of the device. Although we did not focus on power usage in our study, we found that our smart phone would lose 5-10% of its battery per hour of active scanning, which is prohibitive for casual users. It may be possible to extend the battery life of the device by increasing the ϵ parameter of our topological classifier long enough so that scanning can be conducted periodically. Our system would also benefit from a more efficient diameter calculation algorithm as well as a principled method for retiring stale data. We defer these improvements to future work.

VII. Conclusion

We introduced BLE-Doubt, an open-source, smartphone-based BLE counter-surveillance application which rapidly detects nearby Bluetooth trackers and determines whether or not they may be malicious. BLE-Doubt uses a novel topological algorithm which classifies BLE devices as suspicious before notifying the user of their presence. We compared our algorithm to baseline trajectory statistics using a novel validation dataset collected by the authors. Our analysis showed that BLE-Doubt detects real threats about as well as the baselines, but with a vastly decreased prevalence of false positives, and an unmatched F_1 statistic of 0.93. We proposed manufacturer-side improvements to trackers which may help them resist exploitation ranging from changing the Bluetooth Standard to making trackers more physically visible.

Acknowledgements

We thank the Mapbox Community Team for providing free/discounted access to the Mapbox Mobile SDK through January 2022 for the BLE-Doubt app. We also thank Professor David Kohlbrenner for his consultation.

REFERENCES

[1] BITTNER, N., AND HEINRICH, A. AirGuard - AirTag tracking protection. https://github.com/seemoo-lab/AirGuard, 2022. Accessed: 2022-1-18.

[2] BOWLES, N. Thermostats, Locks and Lights: Digital Tools of Domestic Abuse. *The New York Times* (2018). https://www.nytimes.com/2018/06/23/technology/smart-home-devices-domestic-abuse.html.

[3] CHANG, K.-H. Bluetooth: a viable solution for IoT? [Industry Perspectives]. *IEEE Wireless Communications 21*, 6 (2014), 6–7.

[4] CHINCHOR, N., AND SUNDHEIM, B. M. Muc-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993* (1993).

[5] CLAYTON, J., AND DYER, J. Apple AirTags - 'A perfect tool for stalking'. *BBC* (2022). https://www.bbc.com/news/technology-60004257.

[6] FOWLER, J. A. Apple's AirTag trackers made it frighteningly easy to 'stalk' me in a test. *The Washington Post* (2021). https://www.washingtonpost.com/technology/2021/05/05/apple-airtags-stalking/.

[7] FRASER, C., OLSEN, E., LEE, K., SOUTHWORTH, C., AND TUCKER, S. The new age of stalking: Technological implications for stalking. *Juvenile and Family Court Journal 61*, 4 (2010), 39–55.

[8] FREED, D., PALMER, J., MINCHALA, D., LEVY, K., RISTENPART, T., AND DELL, N. A Stalker's Paradise: How Intimate Partner Abusers Exploit Technology. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18* (2018), ACM Press, pp. 1–13.

[9] HOCKING, J., AND YOUNG, G. S. Topology, (1961). *Addison-Wesley, Reading. MR 23* (1988), A2857.

[10] INC., A. Airtag. https://www.apple.com/airtag/, 2021. Accessed: 2021-5-12.

[11] INC., A. Apple Introduces AirTag. https://www.apple.com/newsroom/2021/04/apple-introduces-airtag/, 2021. Accessed: 2022-1-20.

[12] INC., A. Tracker Detect. https://play.google.com/store/apps/details?id=com.apple.trackerdetect, 2022. Accessed: 2022-1-18.

[13] LEVY, K., AND SCHNEIER, B. Privacy threats in intimate relationships. *Journal of Cybersecurity 6*, 1.

[14] MAC, R., AND HILL, K. Are Apple AirTags Being Used to Track People and Steal Cars. *The New York Times* (2021). https://www.nytimes.com/2021/12/30/technology/apple-airtags-tracking-stalking.html.

[15] MALANDAIN, G., AND BOISSONNAT, J.-D. Computing the diameter of a point set. *International Journal of Computational Geometry & Applications 12*, 06 (2002), 489–509.

[16] MATTHEWS, T., O'LEARY, K., TURNER, A., SLEEPER, M., WOELFER, J. P., SHELTON, M., MANTHORNE, C., CHURCHILL, E. F., AND CONSOLVO, S. Stories from survivors: Privacy & security practices when coping with intimate partner abuse. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), pp. 2189–2201.

[17] MCGUIRE, P. Should We Be Freaked Out About Tile, the World's Cheapest New Location Tracker? *Vice* (2013). https://www.vice.com/en$_us/article/qbedmx/should-we-be-freaked-out-about-tile-the-worlds-cheapest-new-location-tracker$.

[18] NETWORKS, R. Android beacon library. https://altbeacon.github.io/android-beacon-library/, 2019. (Accessed on 05/30/2021).

[19] RÖBESAAT, J., ZHANG, P., ABDELAAL, M., AND THEEL, O. An improved BLE indoor localization with Kalman-based fusion: An experimental study. *Sensors 17*, 5 (2017), 951.

[20] ROBINS, V., MEISS, J. D., AND BRADLEY, E. Computing connectedness: Disconnectedness and discreteness. *Physica D: Nonlinear Phenomena 139*, 3-4 (2000), 276–300.

[21] RYU, G.-S. BLE Beacon Based Online Offline Tourism and Solutions for Regional Tourism Activation. *Journal of The Korea Internet of Things Society 2*, 2 (2016), 21–26.

[22] SEMICONDUCTOR, N. nRF Connect for Mobile. https://www.nordicsemi.com/Software-and-tools/Development-Tools/nRF-Connect-for-mobile, 2021. Accessed: 2021-5-13.

[23] SLUPSKA, J. Safe at home: Towards a feminist critique of cybersecurity. *St Antony's International Review 15*, 1 (2019), 83–100.

[24] TEAM, T. Battery Life - Tile Support. https://tileteam.zendesk.com/hc/en-us/articles/360023759353-Battery-Life-, 2021. Accessed: 2021-5-12.

[25] TSENG, E., BELLINI, R., MCDONALD, N., DANOS, M., GREENSTADT, R., MCCOY, D., DELL, N., AND RISTENPART, T. The tools and tactics used in intimate partner surveillance: An analysis of online infidelity forums. In *29th {USENIX} Security Symposium ({USENIX} Security 20)* (2020), pp. 1893–1909.

[26] VELAZCO, C. How to search for Apple's helpful — and unsettling — AirTag trackers - The Washington Post. *The Washington Post* (2021). https://www.washingtonpost.com/technology/2021/12/21/how-to-find-airtags-android-app/.

[27] WOOLLEY, M. Bluetooth technnology protecting your privacy. https://www.bluetooth.com/blog/bluetooth-technology-protecting-your-privacy/, 2015. Accessed: 2021-1-26.

[28] WOOLLEY, M. Bluetooth core specification v5., 2019.