Track You: A Deep Dive into Safety Alerts for Apple AirTags

Narmeen Shafqat
Northeastern University
shafqat.n@northeastern.edu

Nicole Gerzon
Northeastern University
gerzon.n@northeastern.edu

Maggie Van Nortwick
Northeastern University
vannortwick.m@northeastern.edu

Victor Sun
Northeastern University
sun.v@northeastern.edu

Alan Mislove
Northeastern University
amislove@ccs.neu.edu

Aanjhan Ranganathan
Northeastern University
aanjhan@northeastern.edu

ABSTRACT

Bluetooth-based item trackers have sparked apprehension over their potential misuse in harmful stalking and privacy violations. In response, manufacturers have implemented safety alerts to notify victims of extended tracking by unknown item trackers. In this study, we specifically investigate the anti-stalking mechanism of Apple's AirTag. We identify and analyze potential triggers of safety alerts that have not been examined in previous research, such as the local time, the victim's device model, AirTag's battery life, and the distance between the AirTag and the victim's device. Furthermore, we demonstrate a novel possibility of developing a stealthy cloned AirTag capable of tracking victims directly on the Find My app while circumventing safety alerts on the victim's device. Our experiments demonstrate that, despite regular updates to the public key and MAC address, our cloned AirTag can provide real-time location updates even with a four months old key, thereby highlighting the challenges in designing a robust anti-stalking framework. Furthermore, we propose practical solutions to mitigate stalking risks from cloned AirTags and enhance the existing anti-stalking safeguards for AirTags. These suggestions seek to provide a foundation for similar Bluetooth-based item trackers to improve their anti-stalking protections while ensuring optimal tracking efficiency. We conducted rigorous experiments to validate our findings, ensuring their accuracy and reliability. Our evaluation highlights that safety alerts take over 8 hours to appear during the day and are more prompt during the night, particularly after 11 pm.

KEYWORDS

AirTags, Apple, Find My, Anti-stalking, Safety Alerts, Item Tracker

1 INTRODUCTION

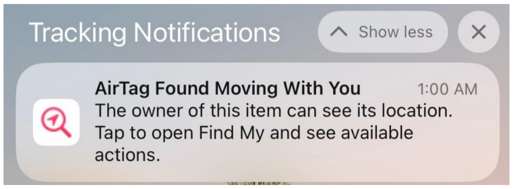
Item trackers such as Apple AirTags [6], Samsung Galaxy SmartTags [43], and Tile Mate [46] are small, affordable, battery-operated devices that can be attached to valuable items such as keys, wallets, cameras, and luggage to make them trackable. These devices are paired with a smart app that leverages Bluetooth Low Energy (BLE) [36] technology to help users easily locate their belongings. If an item tracker is not in range of the owner device, it relies on nearby smartphones (*finder devices*) to hear the BLE advertisements and update its location to a central server, enabling the owner to determine the tracker's location. The precision and effectiveness of an item tracker depend on the size of its finder network, as well as the number of participating devices in close proximity [32]. Since 2013, Tile had established itself as the market leader, with over 35 million trackers in circulation [47]. However, the introduction of AirTags brought about a profound paradigm shift as they leveraged the seamless integration of Apple's *Find My* network with the native *Find My* app [5], which has been pre-installed on all Apple devices since 2015. This resulted in a massive finder network of almost 1.2 billion devices [15], empowering AirTags with unmatched location precision and tracking capabilities.

Unfortunately, AirTags have also brought severe privacy concerns to the forefront, with more than 150 reported cases of unwanted tracking, stalking, and theft of high-end cars [16]. In most cases, victims were unknowingly tracked, harassed, threatened, or even murdered [33] by their estranged partners or spouses who had secretly placed the AirTag in their belongings [3, 9, 13]. These incidents have prompted two victims to file a class-action lawsuit against Apple [4]. Digital privacy experts, tech safety watchdogs, and digital rights advocates worldwide have expressed concerns about the potential misuse of AirTags for stalking or human trafficking and raised questions about who would be liable if the technology is misused; the stalker or the manufacturer. These privacy concerns are exacerbated by other major smartphone manufacturers, such as Samsung and Google, who are following Apple's approach to BLE item trackers. For instance, Samsung has introduced SmartTags, which can be tracked using Samsung's Galaxy Find Network [39] and the SmartThings Find app [29] that is pre-installed on 0.2 billion devices [49]. Reportedly, Google is also developing its tracker tags that are projected to have an even more extensive network of over 3 billion devices [41]. As a result, these privacy concerns are only expected to escalate. Technology companies must take proactive measures to ensure that their products are not used to violate individuals' privacy and safety.

While item trackers operate within their unique finder networks and may differ in their implementation, they share the common fundamental functionality of locating lost items and are subject to similar privacy and security concerns. Given Apple's current market dominance, this paper explores privacy concerns related to BLE item trackers using Apple AirTags as a prime example. To address stalking concerns, Apple has incorporated various anti-stalking measures in iPhones, such as sending a time-sensitive *safety alert* to the victim's device when it detects an unknown AirTag moving along for an extended time (refer to Figure 1). The victim can locate the AirTag using Precision Finding or by playing

Proceedings on Privacy Enhancing Technologies 2023(4), 132–148

https://doi.org/10.56553/popets-2023-0102



Tracking Notifications

AirTag Found Moving With You 1:00 AM
The owner of this item can see its location. Tap to open Find My and see available actions.

Figure 1: Time-sensitive Safety Alert generated by victim's device due to prolonged detection of an unknown AirTag.

a sound. However, such protections have limitations. For instance, if the Bluetooth of the victim's device is turned off, the device will not generate any alert. Also, to prevent false alarms for individuals close to the victim, safety alerts are typically triggered if the victim has been stalked for several hours or upon arrival at their home (or frequently visited location) with that unknown AirTag [27]. Unfortunately, the victim's device generates the safety alert after few minutes of reaching home. By that time, the stalker may have already become aware of the victim's location, creating a significant security and privacy risk. Additionally, proactive safety alerts are not available to Android users. Although Apple released an Android application called "Tracker Detect" in December 2021 [7], it is highly insufficient, as the victim must conduct a manual scan to detect if an AirTag has been tracking them for a prolonged period.

Existing literature provides limited insights regarding safety alerts, primarily focusing on determining the minimal exposure time and distance required to trigger them [38]. Consequently, additional research is needed to fully understand the scope and effectiveness of Apple's anti-stalking measures. In addition, the development of OpenHaystack [24], an open-source framework that enables stalkers to create fake AirTags and exploit *Find My* services highlights the urgent need for action. As finder iPhones are unable to distinguish between genuine and fake AirTags, they keep uploading location reports to the Apple server. This poses a persistent stalking risk until Apple devises effective measures to block fake AirTags. Several researchers have also demonstrated the feasibility of circumventing safety alerts by modifying specific bytes in the BLE payload [38], disabling the speaker [10], and periodically broadcasting new public keys [19] to avoid detection by the victim's device. These works reveal deficiencies in Apple's anti-stalking measures that need addressing to provide improved protection against stalking. In contrast to AirTags, other BLE item trackers even lack this essential proactive safety alert feature, rendering them more vulnerable to stalking.

Our study aims to answer the following research questions:

(1) How do safety alerts work within the AirTag's ecosystem?
(2) What are the limitations of the AirTag's ecosystem in effectively preventing stalking incidents?
(3) What are the challenges associated with building a robust anti-stalking framework that effectively safeguards users' privacy, ensures efficient tracking functionality, and maintains wide applicability across different BLE item trackers?

Our study thoroughly evaluates the effectiveness of safety alerts, providing an unprecedented examination of the circumstances that trigger them, including the time of the day, the model and screen status of the victim's device, the mode and battery life of the AirTag, the victim's mode of transportation, the density of people in the vicinity, the distance between the AirTag and the victim's device, and whether the victim visited any "*Significant locations*" while being stalked. Our evaluation highlights that safety alerts take more than 8 hours to appear during day time, and are more prompt during late-night hours, particularly after 11 pm. Moreover, if the victim has activated the *Significant Location* feature on their device [27], which tracks frequently visited places, such as their home or workplace, safety alerts are activated within 10 minutes of arriving at such location. Based on the insights from the performance evaluation, we aim to identify any potential weaknesses that stalkers could exploit to circumvent safety alerts. To accomplish this, we developed a stealthy cloned AirTag that utilizes an ESP32 controller to broadcast the public key of a genuine AirTag. Unlike existing fake AirTags, our cloned AirTag allows uninterrupted tracking of victims directly through the *Find My* app, without relying on the OpenHaystack Framework. Our analysis of BLE advertisement frames revealed previously undisclosed bits that prevent safety alerts for unfamiliar AirTags on the victim's device, regardless of the distance traveled or exposure time. With this, we demonstrate that despite regular updates to public keys and MAC addresses, our stealthy cloned AirTag can stalk victims on the *Find My* app even with a four months old key, thereby exposing significant weakness in the Airtag's ecosystem. This occurs because the victim's device cannot distinguish between a genuine AirTag (which changes MAC address and public key daily) and a cloned AirTag (with unchanging attributes), resulting in continuous location updates on the server. Furthermore, the study provides practical solutions for improving existing anti-tracking deterrents and preventing the misuse of cloned AirTags, ultimately reducing the possibility of harmful stalking. Our recommendations can help manufacturers of other BLE item trackers implement effective anti-stalking measures.

Specifically, this paper presents the following contributions:

(1) We thoroughly analyze Apple's safety alerts to gain insights into the functioning of the safety alert system and experimentally determine the conditions under which alerts are displayed to the user, such as the victim's location, mode of transportation, local time, and battery level of the AirTag.
(2) We demonstrate how a stalker can use a stealthy cloned AirTag to track a victim on Apple's *Find My* app indefinitely, for a period of four months, without triggering any safety alert on the victim's device. This highlights potential weaknesses in AirTag's ecosystem.
(3) We provide recommendations to improve anti-stalking measures for AirTags and other BLE item trackers. This includes slight modifications at the device and server to block cloned AirTags, validate public keys, enforce limits on the lifespan of these public keys, and offer an optional aggressive scanning feature for at-risk individuals.

2 BACKGROUND AND RELATED WORK

This section explains the functionality of AirTags, their BLE advertisement structure, and the anti-stalking measures implemented by Apple. It also covers relevant research conducted on this subject.

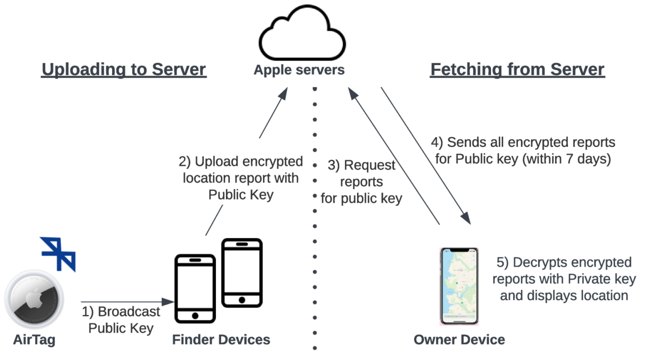

Uploading to Server
Apple servers
Fetching from Server
2) Upload encrypted location report with Public Key
3) Request reports for public key
4) Sends all encrypted reports for Public key (within 7 days)
5) Decrypts encrypted reports with Private key and displays location
1) Broadcast Public Key
AirTag
Finder Devices
Owner Device

Figure 2: Overview of how the Offline Finding (OF) protocol enables long-range location tracking for Apple AirTags.

2.1 How AirTags Work

AirTags are offline item-tracking devices that do not have GPS, Wi-Fi, or cellular connectivity and utilize BLE for communication. AirTags are compatible with any Apple device running iOS/ iPadOS 14.5 or later, including iPhones, iPads, and iPod Touch. To locate an item, the AirTag is paired with the owner device through the *Find My* app. The pairing process links the AirTag's unique serial number to the owner's Apple ID and generates an elliptic curve P-224 public-private key pair and a random secret. This pairing information is securely stored on Apple's backend for up to 25 days [25]. The random secret is used to derive an infinite number of rotating key pairs that are synchronized across all owner devices signed in to the same Apple account via the iCloud keychain.

When the AirTag is within BLE range of the owner device, it can be located directly on the *Find My* app. However, when the owner device is not in proximity, the AirTag relies on Apple's crowd-sourcing network for location updates. The Offline Finding (OF) protocol, illustrated in Figure 2, outlines how these offline devices are tracked within the *Find My* ecosystem. The AirTag periodically broadcasts BLE advertisements containing a public key every 2 seconds. The public key is regularly rotated to prevent user tracking through these advertisements [31]. When finder devices detect the BLE advertisements, they upload location reports encrypted with the advertised public key along with its hash to the Apple server using HTTPS POST request [24]. These reports contain the current location, an estimate of location accuracy, the time the advertisement was received, and the attempted upload time. Each request is authenticated to ensure that only legitimate Apple devices can upload reports to the server. In order to retrieve the location reports, the owner device initiates an HTTPS POST request to the Apple server, authenticating itself using the associated Apple ID. This process generates a tokenized one-time password called AnisetteData, which is then used by the device to request the location reports for the most recent public keys of the AirTag [24]. In response, the Apple server provides all the reports associated with the requested keys. The owner device decrypts these obtained reports using the corresponding private key and displays the AirTag's recent location on the *Find My* app. In short, Apple protects privacy by incorporating end-to-end encryption for location reports, frequent rotation of public-private key pairs to prevent user tracking, and utilizing anonymous location updates sourced from nearby finder devices.

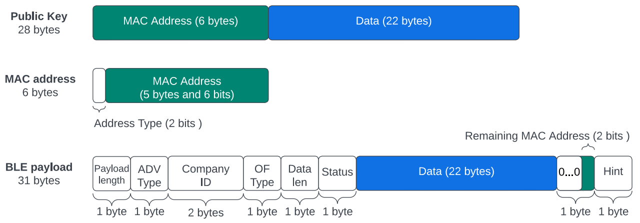


Figure 3: Encoding 28 bytes AirTag's public key in 37 bytes BLE Advertisement packet.

2.2 AirTag's BLE Advertisement

The standard BLE advertisement has a maximum size of 37 bytes, of which 6 bytes are reserved for the MAC address and 31 bytes for the payload [35]. However, to fit the 28-byte public key of the AirTag in the BLE advertisement, Apple stores the first 6 bytes of the key in the MAC address field and the remaining 22 bytes in the manufacturer data field, as illustrated in Figure 3. To identify the MAC address as a random static address (that does not require registration with IEEE), the first two bits of the MAC address field are set to 0b11. As a result, the first two bits of the public key are stored in the byte 29th of the BLE payload.

In addition, the BLE advertisement also includes other fields such as payload length, data length, status byte, data type, OF type, and hint byte. For AirTags, the data type is 0xFF (manufacturer-specific), and the company ID is 0x004C (Apple). The Offline Finding (OF) type indicates the type of service requested by offline devices, e.g., 0x12 for *Find My* services or 0x07 when the AirTag is unpaired. The status byte encodes the device type (e.g., Apple device[1], *Find My* device[2], AirTag, or AirPod) and its battery level (i.e., full, medium, low or critically low). For instance, a status byte value of 0x10 represents a fully charged AirTag, while 0x50 signifies a medium battery level. The hint byte changes every 15 minutes but is not part of the public key. Therefore, the public key of an AirTag remains unchanged for almost 24 hours until its MAC address and data bytes are modified. Finally, a CRC field is included in the BLE advertisement to ensure the packet's integrity.

2.3 AirTag's Operational States

The AirTag operates in four primary internal states; unpaired, connected, nearby, and disconnected [26]. These states are determined by the AirTag's connectivity and distance from its owner device and dictate the content and frequency of its broadcasts.

2.3.1 Unpaired: This is the AirTag's default setting before it is paired with an owner device or after it is removed from a paired account. In this state, the AirTag broadcasts BLE advertisements containing its default MAC address and a portion of its hashed serial number every 33 milliseconds, inviting other *Find My* compatible devices to pair with it.

[1] Any device with a screen, such as iPhones, MacBooks, and iPads.
[2] A third-party *Find My* compatible device, such as Chipolo Spot ONE.

2.3.2 Connected: When the AirTag is within the BLE range of the owner device, which consistently advertises a BLE beacon every 2 seconds, the AirTag enters the connected state and refrains from advertising its public key.

2.3.3 Nearby: This short-lived state (lasting a few seconds to 15 minutes) starts immediately after the AirTag loses connection with the owner device. The AirTag transmits an incomplete BLE packet (i.e., part of the public key stored in the MAC address field) every 2 seconds. This incomplete advertisement prevents finder devices from uploading the AirTag's location to the server while the AirTag remains searching for its owner, who is presumed to be nearby.

2.3.4 Disconnected: Beyond the nearby state, the AirTag transitions into the disconnected state. In this state, the AirTag transmits complete BLE payload every 2 seconds, enabling finder devices to update the location on the central server.

If an AirTag is disconnected from its owner device for more than three days, or the owner device manually sets the tag as lost, it enters lost mode. The lost mode allows supported devices to access a preset "Lost Message" to help reunite the AirTag with its owner. The AirTag transitions back to the connected state when the owner device is back in proximity.

2.4 Apple's Anti-Stalking Protections

The use of item trackers can pose privacy and safety risks if individuals are tracked without their knowledge or consent. To address this issue, Apple has implemented several anti-stalking features into AirTag and iPhone software (beyond i0S 14.5) [8]:

- Safety Alert: Finder devices with Bluetooth, *Find My*, and Location Services enabled can listen to BLE beacons in the background. If an AirTag or *Find My* supported tracker is found moving with the victim's device, a time-sensitive safety alert; *"AirTag Found Moving With You"* is generated on the victim's device (refer to Figure 1).
- Precision Finding: Utilizing Ultra-Wideband (UWB) technology, iPhone 11 and above models can guide the victim to locate the unknown reported AirTag.
- Playing Sound Alert: The victim can locate the unknown reported AirTag by playing a sound.
- NFC Scan: The victim can scan the suspected AirTag with an iPhone or any Near Field Communication (NFC)-capable device to disclose its serial number and the last four digits of the linked phone number (registered with the Apple ID).
- Owner's Information: Apple can disclose the stalker's account details to the victim with a valid subpoena or law enforcement request.
- Disabling AirTag: The victim can remove the AirTag's battery to stop unwanted tracking.
- Automatic Sound Alert: To improve deterrence, an AirTag separated from its owner device for a prolonged time automatically plays a chirp for a few seconds when moved.

To reduce the likelihood of false positives, safety alerts are typically triggered when the victim arrives at a *Significant Location* [27]. With the release of iOS 14.5, Significant Locations are enabled by default, allowing devices to track and store these frequent locations.

Unlike iOS devices, Android devices do not have built-in anti-stalking protections for AirTags. Therefore, Apple has released a proprietary Android app, *Tracker Detect* [7], that allows users to manually search for unknown AirTags in their surroundings. However, its limited functionality and usability are evident from a low rating of 2.3 stars on the Google Play store.

2.5 Related Work

2.5.1 Apple AirTags. Given the limited availability of technical specifications provided by Apple, researchers have endeavored to explore the intricacies of AirTags and associated BLE continuity services [12, 34]. This has involved analyzing AirTag's circuitry [1], extracting firmware through voltage glitching [42], and reverse-engineering Apple's *Find My* (Offline Finding) protocol [24]. The latter resulted in the development of an open-source framework named OpenHaystack, which enables researchers to construct fake AirTags. The stalker generates an elliptic curve P-224 public-private key pair and advertises the public key on a Micro:bit or ESP32 controller. This fake AirTag, emitting similar BLE advertisements like a genuine AirTag, appears legitimate to finder devices, which unknowingly forward the location reports to the Apple server. OpenHaystack framework then queries the server for location reports against the public key and uses the respective private key to decrypt the payload. Since this self-generated key pair is not registered with the Apple ID, the stalker cannot view the location of the fake AirTag on the *Find My* app and instead utilizes OpenHaystack graphical user interface (GUI) [23] or smart app [11] for location updates. In contrast, our cloned AirTags (Section 4) advertise the public key of a genuine AirTag, allowing us to track victims directly on the *Find My* app, without relying on any third-party framework.

Previous research has shown that it is possible to circumvent stalking protections by rapidly changing public keys [19] or by altering the status byte to 0x00, which tricks the victim's device into believing that the transmissions are from an iPhone[3] and does not trigger an alert [38]. We identified additional yet previously undisclosed values of status byte that can also bypass safety alerts.

In practice, Apple's Tracker Detect app can not perform automatic background scans, which limits its ability to provide proactive safety alerts for Android users, especially for fake AirTags. Researchers have used a combination of dynamic and static analysis techniques to reverse-engineer Apple's anti-stalking protections in iOS and discovered that AirTags are considered suspicious after following a user for a minimum of 840 meters and at least 10 minutes [22]. However, to prevent false positives, the alerts are delayed until a specific time. Building upon their findings, the team developed an open-sourced anti-tracking Android app named "AirGuard" which enables Android users to *manually scan* their surroundings for unknown AirTags and Tile trackers [44].

Furthermore, researchers have proposed a privacy-preserving protocol called Blind My to address the limitations of the *Find My* protocol [37]. Blind My introduces partial blind signatures to add additional cryptographic verification to the *Find My* protocol and restricts AirTags to using only a bounded set of keys, thus preventing third-party (fake) AirTags from using *Find My* services.

[3] As iPhones are expensive and have a relatively short battery life, they are deemed unsuitable for long-term stalking.

Table 1: Commercially available BLE Item Trackers.

Item Tracker	Released	Technology Used
Apple AirTags [6]	Apr 2021	BLE, UWB, NFC
Samsung SmartTag [43]	Jan 2021	BLE UWB
Chipolo Plus [14]	Aug 2016	BLE
Tile [46]	July 2013	BLE,
ProTag Duet [30]	Apr 2014	BLE
Nut Find3 [40]	Nov 2013	BLE

Although the proposed server's operations incur low overhead, it is also important to prevent finder devices from uploading bogus location reports for fake or cloned AirTags.

2.5.2 BLE Item Trackers. Commercially available BLE item trackers, listed in Table 1, can be broadly categorized into two groups: 1) those that are natively compatible with major smartphone manufacturers like Apple AirTag, Samsung Galaxy SmartTag [43], and Chipolo Plus [14], and 2) third-party trackers like Tile Mate and Tile Pro [46], Protag Duet [30], and Nut Find3 [40]. Several researchers have conducted privacy analysis of popular item trackers [48], particularly Tile trackers and Samsung Galaxy SmartTags [49] to uncover any potential weaknesses that could enable stalking. However, these studies do not address proactive safety alerts since these trackers do not include such a feature.

The current state of anti-stalking protections for item trackers is concerning, as there is no standardized approach, and each vendor has implemented their own safeguards. For instance, Tile trackers were launched in 2013 without any anti-stalking safeguards, and it was not until 2021, following AirTag's backlash on stalking incidents, that Tile introduced its "Scan and Secure" feature [32]. This feature allows users to manually scan and detect unfamiliar Tile trackers in the surrounding area. However, to confirm the presence of an unfamiliar tracker, the tracker-associated app recommends the user isolate themselves and move for at least 10 minutes. This approach often leads to high false positives, causing unnecessary panic, particularly in crowded areas. In an effort to combat theft, Tile recently added an anti-theft mode to its trackers, rendering them undetectable by the "Scan and Secure" feature. This prevents thieves from knowing that a Tile tracker is nearby [45]. However, it defeats Tile's anti-stalking protection by making it easier for stalkers to stay anonymous. In order to address both stalking and theft concerns, Tile enables anti-theft mode only after the user provides a valid government-issued ID and agrees to stringent usage terms, such as a $1 million penalty for misusing the tracker that could result in a court conviction. In contrast, for Samsung SmartTags, the safety alerts are only initiated on Android devices if they detect an unknown SmartTag in proximity for more than 24 hours. This solution may not be practical if the stalker is someone close to the victim, such as a roommate or partner. In summary, it is imperative to standardize anti-stalking features in all BLE-based item trackers, prioritizing user safety and privacy.

Recently, Apple and Google have issued a draft of an industry specification titled "Detecting Unwanted Location Trackers" [28], which outlines guidelines for manufacturers of BLE item trackers to ensure compatibility with unwanted tracking detection technologies across different smartphones. The aim is to enable users to detect, alert, and disable any unknown BLE item tracker, regardless of its manufacturer or smartphone platform. However, the specification merely extends AirTag-inspired anti-tracking technology to other trackers, such as utilizing sound or BLE/NFC to locate the trackers and retrieve the serial number. It cannot detect and block OpenHaystack-based fake AirTags or our stealthy cloned AirTags, ultimately falling short in its goal to fully prevent stalking.

3 EXPERIMENTAL ANALYSIS OF AIRTAG SAFETY ALERTS

With Apple's anti-stalking protections in place, any victim's device that detects an unfamiliar AirTag for a prolonged period generates a time-sensitive safety alert. However, there are inconsistencies in the available literature regarding the exact circumstances in which these alerts are generated on the victim's device. One study discovered that the alert is triggered when an AirTag moves with the victim's device over a distance of one mile [38]. Another study reverse-engineered Apple's anti-stalking protections in iOS and uncovered that the victim's device marks the AirTag as suspicious if it detects an unknown AirTag for more than 840 meters (0.52 miles) and 10 minutes. However, safety alerts are delayed (staged) to a certain time, and researchers were unable to identify the trigger that generates them later on [22]. Therefore, through this work, we perform a comprehensive evaluation of Apple's anti-stalking measures to determine the specific situations in which safety alerts are triggered, providing insights into the functionality of this feature.

3.1 Evaluation Scenario and Setup

3.1.1 Real-world Scenario. We consider a practical scenario where a "stalker" uses an AirTag to covertly track the "victim" through the *Find My* app on their iPhone. To conceal their identity, the stalker registers the AirTag with an anonymous Apple ID, thereby mirroring recent stalking incidents involving AirTags [9, 13]. It is assumed that the stalker has an opportunity to hide the AirTag in the victim's belongings (e.g., bag or car) while the victim has access to a smartphone (iPhone or Android device) to detect any unfamiliar AirTag trailing them.

3.1.2 Experimental Setup. Our device set for these experiments consisted of 4 iPhones (iPhone XR, iPhone 13 Pro, and 2 iPhone 13 mini), 1 Android device (LG V40 ThinQ), and 4 AirTags. We set up all iPhones with different Apple IDs. An iPhone 13 Mini acted as the stalker's device and was registered with all four AirTags. The remaining iPhones, designated as victim's devices, were expected to generate safety alerts as a result of the prolonged presence of unknown AirTags in their vicinity. From the stalker's perspective, these devices acted as finder devices that would upload the whereabouts of the disconnected AirTags to the Apple server, enabling them to stalk the victim covertly. The Android phone was equipped with Apple's proprietary Tracker Detect [7] app and third-party *AirGuard* app [44] to help detect unfamiliar AirTags.

In all experiments, the stalker's device remained at a fixed location while we, the victims, moved away with the AirTags and smartphones. To ensure accuracy, AirTags were reset and paired with the stalker's device before each experiment and Bluetooth and cellular data were enabled right at the beginning of the experiment (hereby referred to as start time). Consistent conditions were maintained throughout the experiments, including fully charging the devices and AirTags, ensuring devices are functioning normally, and refraining from visiting "significant locations", unless necessary for the experiment. We made deliberate efforts to avoid repeated visits to the same (random) locations to prevent Apple from categorizing them as significant locations. We recorded the time it took for safety alerts to be generated under various conditions and analyzed the difference in alert times in terms of the sample's standard deviation (s). Our results are based on the latest AirTag firmware 2.0.36 (released in Dec 2022) and have been verified for iOS 16.1.2.

3.1.3 Ethical Considerations. It is crucial to emphasize that our study was conducted in an ethical and controlled manner, ensuring no individual was unwillingly or unknowingly subjected to stalking. Our team members assumed the roles of the victim and stalker, eliminating the need for Institutional Review Board (IRB) approval.

3.1.4 Evaluation Metrics. In our experiments, we aimed to investigate the impact of the following parameters on safety alerts:

- iPhone models and capabilities (iPhone XR, iPhone 13 Pro, iPhone 13 Mini, and whether UWB chip is present),
- Local time (day or night),
- Victim's mode of transportation (driving or walking),
- Victim visited any Significant location (e.g., home or workplace) or not during the experiment,
- AirTag's mode (normal or marked lost in the *Find My* app),
- Battery level of AirTags (full, medium, low or critically low),
- Distance between the AirTag and the victim's device,
- Placement of the stalker's device (carried along or not),
- Population density (in the city or in a deserted area),
- Status of the victim's device (screen locked, unlocked, or in Low Power Mode),
- Detection on the Android phone equipped with Tracker Detect and AirGuard App.

3.2 Experimental Approach and Results

Below, we systematically analyze the impact of individual metrics on the generation of safety alerts in order to gain a comprehensive understanding of the specific metrics that activate these alerts.

3.2.1 Victim's Device Models.

Approach: We conducted a three-day experiment to investigate if safety alerts are impacted by different iPhone models (iPhone XR, 13 Pro, and 13 mini) and capabilities (with the latter two equipped with UWB chips and supporting Precision Finding). Starting at 9 am each day, we drove through the city, making intermittent stops, to observe when the safety alert would trigger on the victim's devices and recorded the alert times in Table 7 in Appendix B for reference.

Results: It is crucial to emphasize that throughout the experiments, safety alerts for all four AirTags were triggered simultaneously on each victim's device. Therefore, the recorded alert times in the tables below reflect the alert times for all four AirTags. For this experiment, despite variations in iPhone models and features, all three iPhones generated alerts almost simultaneously every day, with a maximum standard deviation of 1.5 minutes. This indicates that the alert system for detecting unknown AirTags is intended to function uniformly across all models under consideration, provided all devices were exposed to the same AirTags for the same duration.

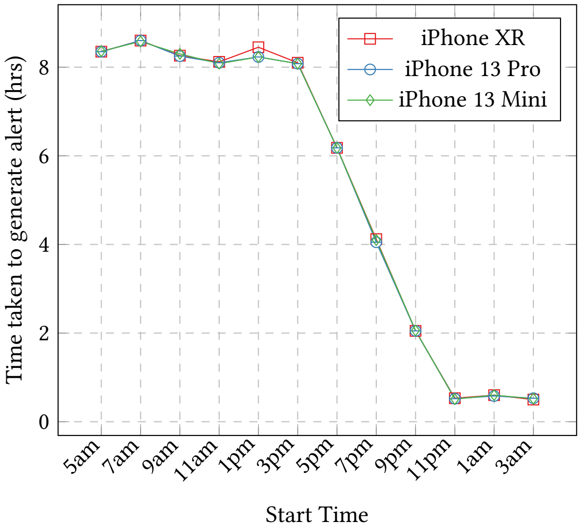


Figure 4: Investigating the Impact of Local Time on Safety Alerts across different phone models.

3.2.2 Local Time.

Approach: In order to investigate whether the local time affects the safety alerts, we carried out a series of experiments over several days, shifting the start time by two-hour intervals each day. The time taken to generate the safety alert was recorded in Table 8 in Appendix B and visually represented in Figure 4.

Results: The findings indicate that it took approximately eight hours for the safety alerts to be triggered during the daytime. The alert time gradually decreased after 3 pm until 11 pm, after which the alerts were triggered within an average of 30 minutes. The faster results obtained during nighttime may be attributed to the heightened risk of stalking being more detrimental during these hours and the lower possibility of false positives due to the lower population density. Another contributing factor might be the periodic update of the MAC address and public key of the disconnected AirTag around 4 am local time [1]. This update results in the victim's device being unable to recognize the AirTag as the same device that it previously detected, potentially leading to prompt safety alerts. Furthermore, we observed that even if the AirTag and the victim's device are placed close for a substantial period prior to driving, the 8-hour timer would start only when the victim starts moving.

We conducted additional experiments starting at 9:30 pm to investigate further whether the safety alerts were expedited after 11 pm. Our findings, presented in Table 9 in Appendix B and visually illustrated in Figure 5, validate that victim's devices do indeed generate safety alerts more quickly beyond this time.

3.2.3 Victim's Mode of Transportation.

Approach: The previous experiments were conducted while driving across the city. To improve our understanding of whether the

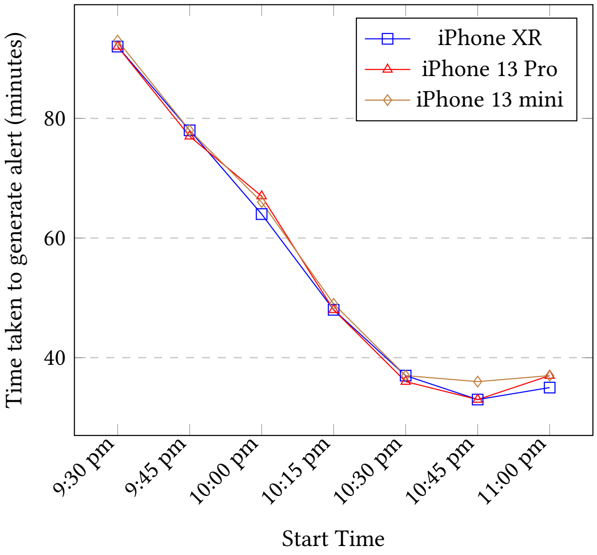


Figure 5: Investigating Accelerated Safety Alerts after 11 pm.

mode of transportation impacts safety alerts, we walked through the city with the victim's devices and AirTags. Knowing that safety alerts are promptly generated during night hours, we conducted subsequent experiments mostly during the night hours. This enabled us to repeat the experiments multiple times, ensuring the validity of the results while effectively managing our time.

Results: Our results, as presented in Table 2, are consistent with the findings from the driving experiment (Table 8) as the victim's device generated the safety alert after 8 hours of walking during daylight hours and after 30 minutes past 11 pm. This suggests that the victim's mode of transportation does not impact safety alerts, as long as the minimum exposure distance and time thresholds are met. Furthermore, we observed that in cases where the victim's device generates the safety alert after midnight, they are reminded about the alert within a time frame of 30 to 40 minutes, regardless of whether the victim's device is stationary or located at home. This continues until either the victim opens the safety alert notification or the public key of the AirTag is changed with its daily rollover.

3.2.4 *Visiting Significant Location.*

Approach: Find My app on iPhone automatically enables the *Significant Locations* feature when location services are activated. In our subsequent driving experiments, we aimed to assess the affect of visiting significant locations on safety alerts. To achieve this, we disabled the *Significant Locations* feature on one of the three victim's devices and cleared the existing history of locations. We then drove for over thirty minutes before returning home and recorded our observations in Table 10 in Appendix B for reference.

Results: The two devices with the activated feature generated the safety alert within 10 minutes of arriving home, irrespective of the time of day. This prompt alert serves as an effective means of alerting the victim of potential stalking risks. It also helps minimize false positives by confirming the presence of an unknown AirTag following the user, rather than a random disconnected AirTag belonging to someone else in a public location. However, when the *Significant Locations* feature was disabled on the victim's device, the device stopped tracking safe locations and did not trigger any safety alert, even when the victim reached home.

Table 2: Investigating the Impact of Mode of Transportation (Walking) on Safety Alerts. The standard deviation of the sample (s) indicates the variation in results in minutes.

Start Time	Time after alert generated			s
	iPhone XR	iPhone 13 Pro	iPhone 13 mini	
9 am	8 hrs 3 mins	8 hrs 6 mins	8 hrs 5 mins	1.53
1 pm	8 hrs 1 mins	8 hrs	8 hrs 1 mins	0.58
9 pm	2 hrs 3 mins	2 hrs 02 mins	2 hrs 05 mins	1.53
11 pm	33 mins	32 mins	32 mins	0.58
1 am	31 mins	31 mins	31 mins	0
3 am	32 mins	31 mins	32 mins	0.58

3.2.5 *AirTag's Mode.*

Approach: Typically, users have the option to mark their misplaced or stolen AirTag as lost in the *Find My* app. To assess whether enabling the lost mode feature generates a safety alert, we marked two out of four AirTags as lost from the stalker's device and performed experiments for three consecutive days starting at 11 pm.

Results: In all trials, irrespective of whether the lost mode is enabled on AirTags or not, safety alerts for all 4 AirTags were consistently generated simultaneously on all victim's devices. This highlights Apple's commitment to maintaining this crucial feature, effectively preventing potential stalkers from exploiting the threat of theft to conceal their stalking activities and ensuring that the victim's device generates alerts even for the supposedly lost AirTag.

3.2.6 *AirTag Battery.*

Approach: We conducted further testing to see if different battery levels of the AirTags affect safety alerts. To accomplish this, we used CR2032 battery cells with varying charge levels (fully charged, medium, low, and critically low) in the AirTags and repeated the experiments for three consecutive days starting at 11 pm.

Results: Regardless of the battery level, all victim's devices generated safety alerts simultaneously for all four unknown AirTags in all three trials. This can be attributed to the fact that the frequency of BLE advertisements emitted by the AirTag remains consistent, even when the battery charge is low. Specifically, when the battery is critically low, the *Find My* app displays a low battery icon and generates a low battery notification on the victim's device.

3.2.7 *Distance between AirTag and the Victim's Device.*

Approach: Next, we attempted to ascertain the range at which a trailing AirTag would trigger a safety alert on the victim's device. Our objective was to evaluate the trade-off employed by Apple in specific scenarios, e.g., detecting concealed AirTags in the victim's vehicle while preventing false alerts for non-victims traveling on the same bus or train. These experiments were again conducted starting 11 pm owing to prompt safety alerts at this time. We placed

AirTags at a fixed point (x) and positioned team members, each holding a victim's device, at varying distances (1, 2, 3, and 4 meters) from the AirTags. We then repeated the experiment twice with a different set of distances as shown in Table 11 in Appendix B and recorded the time when the safety alert was generated.

Results: Our findings indicate that after more than 30 minutes, safety alerts were generated by the victim's devices within a proximity of 4 meters from the unknown AirTag on the bus. However, for passengers who were seated at a greater distance, their devices did not generate safety alerts. This may be one of the reasons why safety alerts are delayed during the day because otherwise, all passenger devices in the bus/ train that are up to 4 meters away from the disconnected AirTag would generate a safety alert, causing unnecessary panic and confusion.

3.2.8 Placement of Stalker's Device.

Approach: In the previous experiments, the stalker's device remained at a fixed location and was not carried along. We conducted additional experiments (post 11 pm) for three consecutive days where the stalker's device, with the Bluetooth turned off and cellular data enabled, was carried alongside the victim's devices. The objective was to determine if the victim's devices would still generate an alert in such a scenario.

Results: Our findings revealed that all victim's devices generated the safety alert after approximately 30 minutes, despite the presence of the stalker's device. Note that this is a false positive scenario that can occur in real-life situations where a benign device owner is traveling in public transport with a disconnected AirTag.

3.2.9 Population Density.

Approach: While evaluating anti-stalking protections in iOS, a previous study indicated that the people density scan is not activated in iOS 15.2 [22]. We conducted two sets of experiments to investigate if finder devices density in the vicinity influences the generation of safety alerts in iOS 16.1.2. In the first set, we visited a deserted beach during the day to determine if the safety alerts are triggered quickly due to the low population density. We walked with one victim's device and unknown disconnected AirTags, gradually increasing the number of victim's devices to three. In the second set of experiments, we visited a sparsely populated road alongside a pond at 3 am and recorded our observations in Table 12.

Results: In the beach scenario, the victim's devices did not generate any safety alerts, even after 7 hours of the experiment. This indicates that the people density scan was not activated during the day. However, during the pond experiment conducted at 3 am, safety alerts were once triggered within 22 minutes. Such a prompt alert can be attributed to the increased risk factor during late-night hours, combined with the low population density in the area.

3.2.10 Status of the Victim's Device.

Approach: We aimed to investigate if safety alerts were triggered differently based on whether the victim's device is locked, unlocked, or in low-power mode. Hence, we set up the three victim's devices such that one was locked, another was unlocked, and the third was in the low-power mode with only 15% battery remaining.

Table 3: Investigating the Impact of Device Status on Safety Alerts. The standard deviation of the sample (s) indicates the variation in results in minutes.

Start	Time after alert generated			s
Time	Unlocked	Locked	Low Power	
1 pm	8 hrs 6 mins	8 hrs 5 mins	8 hrs 10 mins	2.64
7 pm	4 hrs 10 mins	4 hrs 12 mins	4 hrs 12 mins	1.15
11pm	37 mins	37 mins	37 mins	0
1 am	34 mins	34 mins	35 mins	0.58

Results: The results of our study as shown in Table 3 indicate that there was no significant impact of the unlocked screen on the generation of safety alerts. In some instances, the device in low-power mode experienced a delay of up to 5 minutes before generating an alert, but this was not always the case.

3.2.11 Detection on Android Phone.

Approach: So far, we have evaluated the effectiveness of proactive safety alerts on iOS. However, as Android devices lack native anti-stalking capabilities, we relied on Tracker Detect and AirGuard apps to detect unfamiliar AirTags on the Android device. To ensure optimal performance, we granted these apps location access and disabled battery optimization.

Results: These apps function similarly to BLE scanners and displayed all disconnected AirTags discovered through BLE beacons, regardless of whether the AirTag followed the victim for an extended period or was merely present during the scan. Tracker Detect allowed us to locate the AirTags by playing a sound, provided the AirTags were detected for over 10 minutes. In contrast, AirGuard marked the AirTags suspicious after the second scan, given that AirTags moved along for more than 30 minutes.

3.2.12 Summary of Results. To conclude, relying solely on manual scans on Android devices is highly insufficient in ensuring the safety of victims. On the other hand, while iOS's proactive alert feature is commendable, it is also not a foolproof solution for identifying stalking. First, these alerts often come too late, after prolonged exposure to the AirTag, rendering it impossible for the victim to take immediate defensive measures. Secondly, triggering an alert when the victim reaches a significant location, means that the victim's location is exposed to the stalker, even before the victim becomes aware of the AirTag's presence, which is concerning.

4 USING CLONED AIRTAG TO CIRCUMVENT SAFETY ALERTS

This section delves deeper into examining Apple's anti-stalking protections using a cloned AirTag that broadcasts the public key of a genuine AirTag, allowing it to be located on the *Find My* app. We aim to modify BLE advertisements of our cloned AirTag in order to uncover any weaknesses that may compromise the effectiveness of the anti-stalking safeguards. Prior research found that modifying the status byte to 0x00 [38] or specific values that identify the device as an iPhone or Mac, does not activate safety alerts [22]. We confirm that this vulnerability still exists in iOS 16.1.2. By carefully analyzing the structure of BLE advertisements (see Figure 3) and systematically modifying specific bytes in the BLE packet, we found that certain other byte values can enable the cloned AirTag to bypass anti-stalking protections.

4.1 Experimental Scenario and Setup

4.1.1 Threat Scenario. This scenario closely mirrors the one described in Section 3.1.1, except that the stalker now utilizes a cloned AirTag that mimics a disconnected AirTag by continuously broadcasting BLE advertisements every two seconds. Additionally, the cloned AirTag has been altered such that it does not trigger any safety alerts on the victim's device while allowing the stalker to monitor the victim on the *Find My* app uninterruptedly and covertly.

4.1.2 Experimental Setup. In addition to the experimental setup outlined in Section 3.1.2, we utilized a MacBook Pro running MacOS 13.1 Ventura, a wireless Bluetooth sniffer, i.e., *UberTooth One* [20], and ESP32 WROOM controllers [17]. The MacBook was logged in with the same Apple ID as the stalker's iPhone 13 Mini.

To clone an AirTag, we used UberTooth One to capture the BLE advertisements of a genuine AirTag and extracted the MAC address and data field bytes to recreate the public key (by reverse engineering Figure 3). Then we modified the status byte to one of the specific values that does not generate safety alerts (as detailed in Section 4.2). Using BLE GAP API [18], we advertised this public key on an ESP32 controller every 2 seconds. It is worth noting that ESP32 does not have a microphone, UWB, or NFC chip, which means anti-tracking measures, such as sound alerts and precision finding are unavailable to the victim. To render the cloned AirTag self-sufficient, we affixed a battery pack to the controller. Finally, we removed the battery of the genuine AirTag, allowing real-time tracking of the cloned AirTag on the *Find My* app. Note that the *Find My* app allows the owner (stalker) to only view the AirTag's last location. To track the victim's entire route, we relied on the MacOS cache, which continually updates the last location of all AirTags in the items.data file located at \Users\Library\Caches\com.apple.findmy.fmipcore [2]. To recreate the victim's route, our Python script retrieves the location and timestamp for the cloned AirTag from the items.data file each time it is updated and plots the coordinates onto a digital map for real-time surveillance, as illustrated in Figure 6.

4.1.3 Ethical Considerations. In addition to conducting our experiments ethically, we acted responsibly by disclosing our findings to Apple and are currently awaiting their response.

4.2 Experimental Approach and Results

4.2.1 Defeating Safety Alerts. In general, the status byte within the BLE advertisement serves to indicate the device type (such as Apple device, AirTag, AirPod, or compatible *Find My* device e.g., Chipolo One Spot) and its battery level [22]. For instance, the AirTag transmits status bytes 0x10, 0x50, 0x90, and 0xD0 when the battery is full, medium, low, and very low, respectively. Our results in Section 3.2.6 indicate the victim's device generates safety alerts irrespective of the AirTag's battery level. We observed during experimentation that it is actually the Most Significant Nibble (MSN) of the status byte, i.e., the first four bits, that determines whether the alert should be generated. Hence, we conducted a series of experiments to determine which status byte values in the BLE advertisements can help bypass anti-stalking measures.

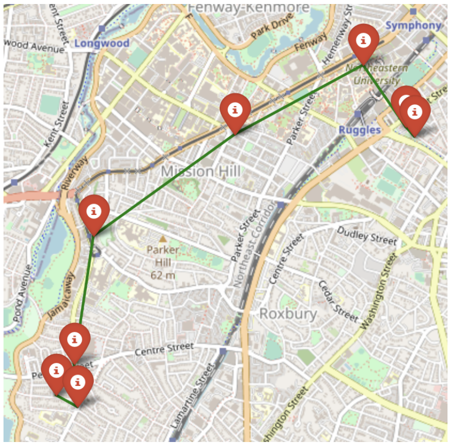


Figure 6: Tracking the tracker: Route of the cloned AirTag retrieved by the stalker using Python script on Mac.

Table 4: Investigating Impact of Changing Status Byte on Detection Efficiency of Find My, Tracker Detect, and AirGuard.

Device Type	Status Byte MSN (hex)	iOS	Android (Manual Scan)	
		Find My Alert	Tracker Detect	AirGuard (30+ mins)
Apple Device	0, 4, 8, C	✗	✗	Apple Device
AirTag	1, 5, 9, D	AirTag	AirTag	AirTag
Find My Device	2, 6, A, E	✗	✗	Find My Device
AirPods	3, 7, B, F	AirPods	✗	AirPods

Approach: We captured public keys for two AirTags, removed their battery, and prepared cloned AirTags on ESP32 controllers. These AirTags transmitted BLE beacons with specific status byte values in each round. We left the stalker's iPhone and MacBook at one location and moved away, as the victim, carrying the remaining smartphones, two genuine, and two cloned AirTags. The genuine AirTags were used as a baseline to determine when the actual alert would appear, while two cloned AirTags to make sure that the victim's device does not generate the alert for any of them. We stayed away from any Significant Location for more than 15 hours (9 am to 12 am). Throughout this time, we checked if safety alerts were triggered on 1) the *Find My* app on Finder iPhones or 2) Tracker Detect or AirGuard app on the Android phone.

Results for iPhone's Proactive Alerts: As indicated in Table 4, the victim's devices only generate a proactive safety alert for the cloned AirTag if the status MSN value corresponds to the device type; AirTags and AirPods. In the latter case, the cloned AirTag mimics an AirPod case, and the victim's device triggers the *AirPod Found Moving With You* alert (as seen in Figure 10, Appendix A). Knowing that device type "Apple Device" does not trigger an alert, our research goes further by demonstrating that transmitting BLE advertisements for "*Find My* device" with status MSN values set to

Table 5: Investigating Lifespan of Cloned AirTag: How long could the stalker locate the cloned AirTag on Find My app?

Time	iPhone's Find My App		Mac's Find My App	
	BLE On	BLE Off	BLE On	BLE Off
Cloned AirTag 1				
Week 1-2	✓	✓	✓	✓
Week 3	✓	✗ (2 days)	✓	✓
Week 4-16	✓	✓	✓	✓
Cloned AirTag 2 (re-cloned after 2 weeks)				
Week 1-14	✓	✓	✓	✓

hexadecimal 2, 6, A, and E also does not generate alerts. We repeated the experiments for these values thrice to validate the results.

Results for Manual scanning on Android Device: The manual scan using Apple's Tracker Detect app could only detect genuine AirTags (i.e., when status MSN values were 1, 5, 9, or D), rendering it incapable of detecting cloned AirTags with other status MSN values. In contrast, AirGuard displayed the MAC address and device type of all BLE-based Apple devices that it detected, using the company ID field (e.g., 0x00FC for Apple) present in the BLE beacons. Consequently, the stealthy cloned AirTag promptly appeared in the device list as a distinct device type matching the status MSN value, even if it was switched on just a few seconds ago and had not moved with the victim. The AirGuard app triggered the safety alert for the specific device type after half an hour of exposure.

4.2.2 *Life of a Cloned AirTag.*

Approach: Our experiments validate the prior discovery that disconnected AirTags effectively prevent unwanted tracking by frequently updating their MAC address and public key, typically around 4 am local time [1]. However, to test whether Apple would identify that our cloned AirTags had not altered its key, we conducted a 16-weeks trial in which our cloned AirTags consistently advertised the same MAC address and public key. We regularly monitored the stalker's *Find My* app to verify that the location for the cloned AirTags was successfully updated.

Results: As indicated in Table 5, the stalker's iPhone successfully received updated location for the cloned AirTags during the initial two weeks. However, for two days in the third week, the iPhone's *Find My* app only updated the location when the device's BLE was turned on and not otherwise. We believed that this occurred because finder iPhones unknowingly update location reports on the server, regardless of the public key's age, but the stalker's device stopped fetching location reports if the AirTag's key had not changed in over two weeks. To investigate this further, we set up a fresh clone of the genuine AirTag using the same approach, which worked without any issues beyond 2 weeks. This indicates that Apple does not impose a limit on the lifespan of public keys. However, the exact technical issue that caused the temporary interruption remains undetermined. Interestingly, after two days, the stalker's iPhone's *Find My* app resumed location updates for the cloned AirTag 1 and has continued to do so, without detection, despite using a 16-week-old public key. In contrast, the *Find My* app on the MacBook did not limit the lifespan of the public key at any time, allowing indefinite tracking without triggering any safety alert. To mitigate the risk of unauthorized and prolonged tracking, we strongly recommend that Apple limits the lifespan of the public key to a maximum of two days. Otherwise, these stealthy AirTags will go undetected by the victim unless they independently discover and remove them.

4.2.3 *Susceptibility to Location Spoofing.*

Approach: Previously, we removed the battery of the genuine AirTag to enable our cloned AirTags to function. Next, we sought to explore the scenario where both genuine and cloned AirTags transmit same public key simultaneously from different locations.

Results: We conducted this experiment thrice and each time the owner's *Find My* app alternated between displaying the location of the genuine AirTag and the cloned AirTag. This happens because finder iPhones cannot differentiate between them, and continue to send location reports to the server. The owner's *Find My* app retrieves the location report from the server and displays the most recent location. It is important to highlight that an adversary can easily capture Bluetooth traffic using a wireless sniffer. They could capture multiple AirTag beacons in a crowded area and then advertise all public keys on ESP32 controllers located elsewhere. This can deceive all AirTag users into falsely believing that their AirTag or the attached item is in a different location for a certain period.

5 MITIGATING THE RISK OF COUNTERFEIT AIRTAGS

In this paper, we categorize counterfeit AirTags into two types: (1) Cloned AirTags, which advertise the public key of a genuine AirTag and can be located on the *Find My* app (Section 4), and (2) Fake AirTags that utilize the OpenHaystack framework and broadcast a self-generated public key (Section 2.5). It is crucial to note that iPhones currently lack the capability to distinguish between genuine, fake, and cloned AirTags, thereby enabling the stalker to exploit *Find My* services. While blocking cloned AirTags provides some level of protection, the absence of serial numbers in fake AirTags poses a significant challenge for Apple to track the stalker, leaving individuals vulnerable to potential stalking incidents. Hence, it is imperative to block both cloned and fake AirTags to ensure comprehensive protection against unwanted tracking. This section presents key insights into how counterfeit AirTags exploit vulnerabilities to access *Find My* services and bypass anti-stalking protections. Consequently, we propose server and device-side approaches to effectively detect and block them.

5.1 Key Insights into the *Find My* Architecture

5.1.1 *Lack of Key Authentication and Verification.*

From a security standpoint, it is crucial for Apple to authenticate both the finder and owner devices to ensure that only genuine Apple devices can upload or fetch location reports to and from the Apple server. However, the current framework lacks mechanisms to verify two essential aspects: 1) the uploaded location reports actually belong to a registered AirTag, and 2) the device requesting the location reports is the legitimate owner of the AirTag. This loophole allows fake AirTags to broadcast self-generated public keys and utilize Apple servers to store and retrieve location information. To gain a deeper understanding, we modified the OpenHaystack code and analyzed real location reports retrieved from Apple's backend. Besides location data, these reports also include additional details such as upload timing and assumed accuracy. It is worth noting that our modified code allowed us to request and receive *encrypted* reports for any public key, including those associated with genuine AirTags registered to other users (see Figure 11 in Appendix C).

Does that imply that Apple can not verify if a certain public key belongs to a registered AirTag and is associated with an Apple account? However, several facts and evidence bolster the notion that Apple maintains an internal log that connects AirTag's serial numbers, public keys, and Apple IDs. For instance, an AirTag (with a unique serial number) can only be registered to a single Apple ID, and all devices registered under that Apple ID have access to the keying information required to locate the AirTag. Additionally, even if the AirTag is reset manually (by removing the battery five times), the owner device must delete the AirTag before it can be paired again. This deletion process requires an active internet connection to communicate with the Apple servers. Moreover, Apple itself states that, with a valid subpoena, they can determine the Apple ID associated with the serial number of the AirTag being used for unwanted tracking [25]. Finally, although fake AirTags can request location reports from the server, they can not display their location on the *Find My* app. These indications suggest that Apple can authenticate and associate public keys with their corresponding Apple IDs. A straightforward solution to prevent fake AirTags from uploading location reports for unregistered keys is to maintain a mapping between the Apple ID and the hash of associated public keys and verify the association. Unlike storing actual public keys, storing their hash does not enable Apple or third parties to track the victim's location. Similarly, the server endpoint responsible for handling fetch requests should also cross-reference this key hash-Apple ID database to authenticate whether the keys used to request location reports are genuinely associated with the initiating device. It's important to note that the key authentication mechanism can effectively block fake AirTags but not cloned ones, underscoring the necessity to explore alternative approaches.

5.1.2 Inconsistencies in AirTag's Detection and Alert Mechanism. As shown in Section 4, finder devices upload location reports to the central server, regardless of the status byte value in the BLE advertisements. This enables the stalker to track the location of their counterfeit AirTag on the *Find My* or OpenHaystack app, depending on the type of public key used. In contrast, the victim's device only triggers a safety alert if the status byte in the BLE advertisement is identified as belonging to either an AirTag or AirPod. It is not generated for other Apple devices, e.g., iPhones or iPads, which are deemed unsuitable for tracking victims owing to their size and low battery life. This inconsistency in the AirTag's detection and alert generation mechanisms allows the stalker to effectively circumvent the anti-stalking protections without being noticed.

5.1.3 Lack of Device Type Validation. Currently, the owner device fetches location reports from the server against a set of public keys and updates the latest location on the *Find My* app, without verifying the device type. This enables the stalker to advertise BLE packets with the status MSN values that do not trigger safety alerts on the victim's device, but still receive updated location on the *Find My* app. Our analysis of items.data file in the MAC cache reveals that the file does store device type along with location coordinates. To prevent the stalker from getting real-time updates of their stealthy counterfeit AirTags, it is important that AirTag's location must only be updated if the device type corresponds to AirTag.

Table 6: Comparing Genuine vs Counterfeit AirTags.

Type	Genuine AirTag	Counterfeit AirTag	
		Cloned	Fake
Key	Registered with Apple	Registered with Apple	Self-generated (unregistered)
Interface	Find My	Find My	OpenHaystack
PDU type	CONN	CONN	NONCONN
CSA	CSA2	CSA1	CSA1
Key/MAC update	Regularly	No	No
Status MSN (hex)	1,5,9,d for detection	2,6,A,E for stealth	0 for stealth

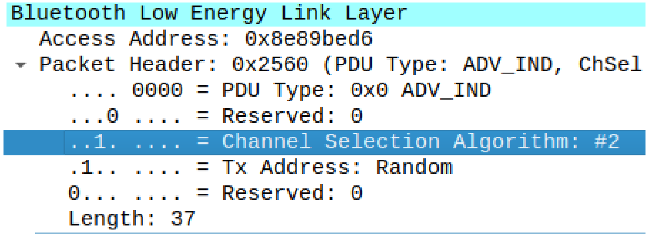
5.2 Device-side Mitigations

Our analysis of BLE advertisements from genuine and counterfeit AirTags identified key differences, summarized in Table 6, that finder devices can leverage to filter out fake BLE advertisements.

5.2.1 Distinct Packet Data Unit (PDU) Type. To conserve battery, Open Haystack-based fake AirTags advertise their public key using Non-connecting Advertising Indicator (ADV_NONCONN_IND), while genuine AirTags (as well as cloned AirTags) broadcast their public key as an Advertising Indicator (ADV_IND) seeking a connection with central devices. While the difference is clear, it is also trivial for an adversary to modify the PDU type of the fake AirTag to transmit ADV_IND packets, making further detection necessary.

5.2.2 Different Channel Selection Algorithms (CSA). CSA aims to identify the most effective channel for transmitting wireless data. When both the BLE master and slave devices are equipped with Bluetooth version 5.0 or higher, CSA 2 is the default channel selected. As a result, when the iPhone (master) and AirTag (slave) communicate, they use CSA 2 as depicted in Figure 7. On the other hand, current counterfeit AirTags developed using ESP32 or other supported hardware only support Bluetooth v4.2 [17], causing them to use CSA 1 for communication. It is worth noting that although the slave device can request the master to change the CSA, it cannot do so independently. Hence, finder devices should reject CSA change requests or disregard AirTag beacons using CSA1. It is important to anticipate stalkers might use hardware for developing counterfeit AirTags that inherently support CSA2 and thus this measure alone is not sufficient to filter fake BLE beacons.

5.2.3 Irregularity in Lifespan of MAC Addresses and Public Keys. AirTags are designed to protect against unauthorized tracking by regularly changing their MAC address and public key. However, counterfeit AirTags use a static key to track the user. In order to prevent unwanted tracking, Apple must restrict the lifespan of the



```
Bluetooth Low Energy Link Layer
  Access Address: 0x8e89bed6
▾ Packet Header: 0x2560 (PDU Type: ADV_IND, ChSel
    .... 0000 = PDU Type: 0x0 ADV_IND
    ...0 .... = Reserved: 0
    ..1. .... = Channel Selection Algorithm: #2
    .1.. .... = Tx Address: Random
    0... .... = Reserved: 0
    Length: 37
```

Figure 7: BLE Link Layer packet header of the beacon transmitted by a genuine AirTag depicting the use of CSA 2.

public key to no more than one or two days. Accordingly, the finder iPhone should cease sending location reports if it detects an AirTag with the same MAC address and public key for more than two days, and alert the user. Similarly, the owner's *Find My* app should stop updating the location of such AirTag to prevent unwanted tracking and force the stalker to reset the AirTag.

To ensure security, it is also crucial to block Find You-like setups that continuously alter public keys every 2 seconds to avoid being flagged as a suspicious AirTag and evade anti-stalking measures [19]. As this irregular behavior is challenging to detect, we suggest that iPhones wait for multiple advertisements (e.g., for 10 seconds) before uploading the first location report for any AirTag.

5.2.4 Generate Safety Alert irrespective of the Status Byte. Addressing the limitation highlighted in Section 5.1.2, the victim's devices should generate a safety alert whenever any device remains in close proximity for an extended period, regardless of the value of the status byte. This will prevent stalkers from bypassing safety alerts.

5.2.5 Validating Device Type before Updating Location on the Find My App. To address the limitation identified in Section 5.1.3, the *Find My* app should only update the location of the AirTag if the device type also corresponds to AirTag.

5.2.6 Blocking Unauthorized Plugins. OpenHaystack utilizes a custom Apple Mail plugin, inheriting entitlements to authenticate as a genuine Apple user and fetch location reports for its self-generated public keys from the server [21]. Apple can implement strict plugin validation to prevent fake AirTags from accessing *Find My* services.

5.3 Server-side Mitigations

While some device-side mitigations can be bypassed by the stalker, we suggest following server-side measures that can effectively prevent stealthy counterfeit AirTags from misusing *Find My* services.

5.3.1 Authenticating Public Keys. Apple servers currently allow uploading and fetching location reports under any public key (whether registered or unregistered). Thus, before storing any location report, the server must validate the mapping between the hash of the report's public key and Apple ID to confirm that the public key is not self-generated and is linked to a registered AirTag. Similarly, when the owner device requests a location report, the server must revalidate the public key's hash and Apple ID mapping to ensure that only the legitimate owner device requests its AirTag's location, and that the request is not originated from the OpenHaystack framework. Although this measure may introduce slight processing time, it plays a crucial role in effectively blocking fake AirTags and significantly enhancing the security of the system.

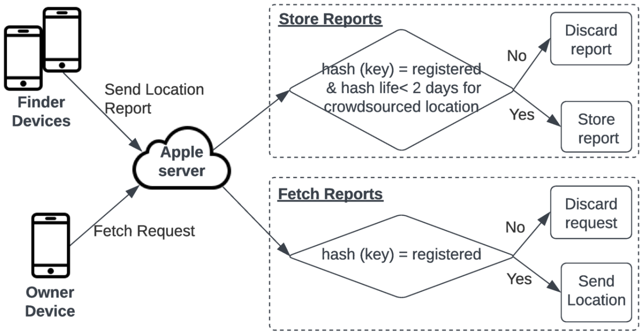


Figure 8: Suggested Server-side Approach to Mitigate the Stalking Risks from Counterfeit AirTags.

5.3.2 Blocking Persistent Public Keys. In general, both fake and cloned AirTags advertise the public key of the AirTag in the *disconnected state*, i.e., the location type for the reports is labeled as "crowdsourced"[4]. Thus, Apple's servers should flag an AirTag as suspicious if its location reports are consistently stored with the same public key hash for more than two days. The servers already have the ability to store location reports for up to seven days, meaning there is an existing infrastructure to filter and identify reports from counterfeit AirTags. We have briefly summarized the server-side mitigations in Figure 8.

6 RECOMMENDATIONS TO IMPROVE ANTI-STALKING MEASURES

The risk of stalking is not limited to AirTags alone, as other BLE item trackers such as SmartTags and Tile can also be misused for unwanted tracking. Unfortunately, the lack of adequate anti-stalking protections in these trackers makes it challenging for victims to identify and counter stalking attempts. This places an onus on tracking device manufacturers to prevent, detect, and proactively inform victims of any stalking activities. However, designing a robust anti-stalking framework that rapidly and accurately identifies an unfamiliar tracker, while also minimizing false alerts to prevent victims from disregarding genuine stalking alerts, poses a significant challenge. Building on our comprehensive analysis of Apple AirTags, we present insightful recommendations to improve anti-stalking safeguards and shed light on the challenges associated with their implementation. This information will be valuable to manufacturers of other BLE item trackers seeking to enhance their anti-stalking protections. We recommend manufacturers prioritize the implementation of the following crucial anti-stalking measures:

Early Proactive Safety Alerts. As victims have limited control over stalking, it is crucial to notify users of any unfamiliar trackers trailing them. Currently, proactive alerts are only generated for unknown AirTags on iOS, while users have to manually scan for

[4] The items.data file in the Mac cache confirms that the reports for the counterfeit AirTags that are sourced from finder devices are termed crowdsourced.

unknown AirTags and SmartTags on Android OS and for unknown Tile trackers on both iOS on Android. Manual scans are ineffective as users are unlikely to actively search for potential threats unless they have suspicions. Moreover, implementing continuous manual scanning at the application level in the background can negatively impact battery life and other Bluetooth-capable features. To tackle this challenge, tracker manufacturers must closely collaborate with smartphone manufacturers to enable background scanning within the OS and facilitate the activation of proactive safety alerts.

Although Apple AirTags offer proactive safety alerts, the prolonged delay of over 8 hours in triggering them is a significant concern, as it undermines the victim's ability to respond promptly to potential threats. This makes it crucial to reduce the alert times substantially. Furthermore, safety alerts should be triggered regardless of the victim's mode of transportation, time of the day, and the battery or operational mode of the victim's device or tracker. The alert should also be generated irrespective of the victim's location to prevent the disclosure of their significant locations to the stalker before they can identify and disable the tracker. These measures can inadvertently increase false positives, where genuine disconnected trackers are mistakenly flagged as suspicious, especially in public transportation or crowded areas. Therefore, a trade-off between prompt alerts and minimizing false positives must be carefully balanced to ensure the effectiveness of the anti-stalking mechanism.

Interoperability. If all manufacturers implement proactive alerts, it becomes possible to achieve interoperability among tracker apps, enabling them to also detect trackers from other manufacturers. This eliminates the need for users to download multiple apps to ensure their security. Recognizing the need for a unified solution, Apple and Google have recently joined forces [28], showcasing their commitment to enhancing users' safety and privacy.

Optional Aggressive Scanning Mode. Manufacturers can consider implementing an optional aggressive scanning mode within the tracker-associated app, allowing potential victims of stalking, such as divorced spouses, celebrities, activists, and journalists in hostile territory, to opt-in for enhanced security. This mode will trigger quick alerts if an unfamiliar tracker remains in close proximity for a period exceeding a predefined threshold, such as 1 hour. Considering the potential for false positives, it is essential to develop methods to accurately distinguish between suspicious trackers and harmless coincidental scenarios to maintain user confidence.

Audible Alerts. Tracker must emit a chirp if it remains separated from the owner device for an extended period, such as 2 days. This serves to alert the victim if the safety alert has gone unnoticed.

Preventing Unauthorized Tracking by External Parties. All location reports must be end-to-end encrypted to prevent unauthorized tracking by the manufacturer or third parties. Additionally, to prevent unwanted tracking using BLE beacons, manufacturers must regularly rotate the tracker's MAC address and public key and ensure identifiers are not reused or used beyond 2 days.

Tracker Sharing. It is important that static users, such as those attending the same class or party, do not receive safety alerts for an unknown disconnected tracker if an attendee (owner) has their device's Bluetooth turned off. Although the anti-stalking algorithm already identifies a stalking attempt based on sustained movement with the tracker, our experiments reveal that alerts are generated even when the owner device has Bluetooth turned off while traveling together. To address this, manufacturers could consider implementing *Tracker Sharing* feature, inspired by Tile trackers. This optional functionality would facilitate collaborative tracking, allowing users to invite trusted individuals, such as their partners or family members, to connect to their item tracker. Consequently, users will not receive unnecessary safety alerts while traveling together or borrowing items with an attached tracker.

Post-detection Measures. Once the unknown tracker is discovered, the user should be able to retrieve its serial number using NFC or BLE technology, enabling them to inquire the manufacturer about the stalker's identity. Like AirTags, users must be able to view the owner's mobile number partially to allow them to verify if the tracker belongs to their partners or friends. Moreover, it should be fairly easy for the user to remove the battery and deactivate the tracker to prevent further location updates. Finally, users must have the provision to report stalking attempts to the manufacturers.

Legal Concerns. With a valid subpoena, the manufacturer must disclose stalker's information to the victim, and cooperate with law enforcement authorities to pursue legal action against the stalker.

User Awareness. Merely displaying a warning during the tracker's registration process, cautioning against misuse, is inadequate in preventing stalking. Manufacturers must educate users about potential stalking risks and the implications of misusing trackers via social media and awareness campaigns.

7 CONCLUSION

Despite their usefulness in helping locate misplaced or lost personal items, BLE item trackers, particularly Apple AirTags, present a significant security threat as they can be used to stalk unsuspecting individuals. This study evaluated the performance of Apple's anti-stalking measures with a focus on identifying previously unexplored circumstances that activate safety alerts, including local time, the victim's device model and screen status, the mode and battery life of the AirTag, the distance between the AirTag and the victim's device, and whether the victim visited any "Significant locations" while being stalked. The study also highlighted the ease with which cloned AirTags can be weaponized, as evidenced by our discovery that changing the status byte to specific values can bypass the anti-stalking measures offered by Apple, making it feasible for the stalker to track the victim even with a four-months-old public key. We proposed several countermeasures to address this issue and provided additional recommendations to harden the anti-stalking protections against unknown AirTags, including adding an optional aggressive scanning mode, authenticating public keys before storing the location reports, and enforcing a limit on the life span of the public keys. These recommendations also serve as a basis for enhancing anti-stalking measures for other BLE item trackers. Our findings were verified through a series of experiments, reinforcing the need for improved security measures in BLE item trackers. Overall, it is crucial to address the security concerns associated with these devices to ensure that they are used responsibly and do not threaten individuals' privacy and safety.

ACKNOWLEDGMENTS

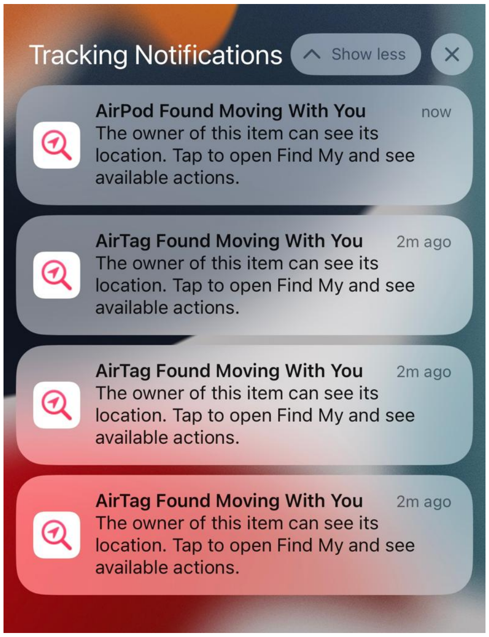
We would like to thank PETS reviewers for their invaluable feedback. This research was partially supported by National Science Foundation under grants numbered 2144914, 1900879, and 1955227.

REFERENCES

[1] Adam Catley. 2022. Apple AirTag Reverse Engineering. Retrieved Nov 2022 from https://adamcatley.com/AirTag.html
[2] Alex. 2022. AirtagAlex. https://github.com/icepick3000/AirtagAlex
[3] Allison McDaniel. 2022. Altered Carbon actor Hannah Rose May shares recent AirTag tracking experience. Retrieved Jan 2023 from https://9to5mac.com/2022/07/05/actress-airtag-tracking-disneyland/
[4] Amanda Holpuch. 2022. Two Women Sue Apple Over AirTag Stalking. Retrieved Jan 2023 from https://www.nytimes.com/2022/12/06/business/apple-airtag-lawsuit.html
[5] Apple Inc. 2017. Find My: One app to find it all. Retrieved Jan 2023 from https://www.apple.com/icloud/find-my/
[6] Apple Inc. 2021. AirTag: Lose your knack for losing things. Retrieved Jan 2023 from https://www.apple.com/airtag/
[7] Apple Inc. 2021. Tracker Detect. Retrieved Jan 2023 from https://play.google.com/store/apps/details?id=com.apple.trackerdetect
[8] Apple Inc. 2022. An update on AirTag and unwanted tracking. Retrieved Jan 2023 from https://www.apple.com/newsroom/2022/02/an-update-on-airtag-and-unwanted-tracking/
[9] Ben Lovejoy. 2022. AirTag stalker imprisoned and given restraining order after tracking ex-girlfriend. Retrieved Jan 2023 from https://9to5mac.com/2022/08/11/airtag-stalker/
[10] Ben Lovejoy. 2022. AirTags with deactivated speakers being sold on eBay and Etsy; seller claims not for stalking. Retrieved Jan 2023 from https://9to5mac.com/2022/02/03/airtags-with-deactivated-speakers-being-sold/
[11] Lukas Burg, Max Granzow, Alexander Heinrich, and Matthias Hollick. 2022. OpenHaystack Mobile-Tracking Custom Find My Accessories on Smartphones. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, New York, NY, 277–279. https://doi.org/10.1145/3507657.3529655
[12] Guillaume Celosia and Mathieu Cunche. 2020. Discontinued privacy: Personal data leaks in Apple Bluetooth-low-energy continuity protocols. *Proceedings on Privacy Enhancing Technologies* 2020 (2020), 26–46. https://doi.org/10.2478/popets-2020-0003
[13] Chance Miller. 2022. AirTag safety features foil Iowa man's repeated stalking attempts. Retrieved Jan 2023 from https://9to5mac.com/2022/12/11/airtag-safety-features-lead-to-arrest-of-stalker/
[14] Chipolo. 2016. Chipolo Plus. Retrieved Jan 2023 from https://chipolo.net/en-us/blogs/new-chipolo-plus-is-here
[15] Daniel Ruby. 2022. 26+ iPhone User and Sales Statistics (Fresh Data 2023).
[16] David Price. 2022. Investigation uncovers dozens of AirTag 'stalking' cases–and that's just the tip of the iceberg. Retrieved Jan 2023 from https://www.vice.com/en/article/y3vj3y/apple-airtags-police-reports-stalking-harassment
[17] Espressif. 2022. ESP32 Series of Modules. Retrieved Jan 2023 from https://www.espressif.com/en/products/modules/esp32
[18] Espressif. 2023. GAP API.
[19] Fabian Braunlein. 2022. Find You: Building a stealth AirTag clone. Retrieved Jan 2023 from https://github.com/positive-security/find-you
[20] Great Scott Gadgets. 2019. Ubertooth One. Retrieved Jan 2023 from https://greatscottgadgets.com/ubertoothone/
[21] Alexander Heinrich. 2021. OpenHaystackMail.
[22] Alexander Heinrich, Niklas Bittner, and Matthias Hollick. 2022. AirGuard-Protecting Android Users from Stalking Attacks by Apple Find My Devices. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, New York, NY, 26–38. https://doi.org/10.1145/3507657.3528546
[23] Alexander Heinrich, Milan Stute, and Matthias Hollick. 2021. OpenHaystack: a framework for tracking personal Bluetooth devices via Apple's massive Find My network. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, Abu Dhabi, UAE, 374–376. https://doi.org/10.1145/3448300.3468251
[24] Alexander Heinrich, Milan Stute, Tim Kornhuber, and Matthias Hollick. 2021. Who can find my devices? security and privacy of Apple's crowd-sourced Bluetooth location tracking system. *Proceedings on Privacy Enhancing Technologies* 2021, 3 (2021), 227–245. https://doi.org/10.2478/popets-2021-0045
[25] Apple Inc. 2021. *Legal Process Guidelines*. Technical Report. Apple Inc, United States.
[26] Apple Inc. 2022. Find My Network Accessory Specification Developer Preview Release R1.
[27] Apple Inc. 2022. What to do if you get an alert that an AirTag, Find My network accessory or set of AirPods is with you.
[28] Apple Inc. 2023. Apple and Google lead initiative for an industry specification to address unwanted tracking.
[29] SmartThings Inc. 2023. SmartThings Find.
[30] Innova Technologies. 2014. ProTag by Duet.
[31] Ivan Krstić. 2019. Behind the Scenes of iOS and Mac Security. In: Black Hat USA 2019. Retrieved Sep 2022 from https://www.youtube.com/watch?v=3byNNUReyvE&t=1893s.
[32] Jared Newman. 2016. To Stand Out In A Sea Of Bluetooth Trackers, Tile Slims Down And Gets Built In. Retrieved Jan 2023 from https://www.fastcompany.com/3063295/to-stand-out-in-a-sea-of-bluetooth-trackers-tile-slims-down-and-gets-built-in
[33] Mark Rasch. 2022. AirTag Stalking – Murder, Fear and Litigation. Retrieved Jan 2023 from https://securityboulevard.com/2022/12/airtag-stalking-murder-fear-and-litigation/
[34] Jeremy Martin, Douglas Alpuche, Kristina Bodeman, Lamont Brown, Ellis Fenske, Lucas Foppe, Travis Mayberry, Erik C Rye, Brandon Sipes, and Sam Teplov. 2019. Handoff all your privacy: A review of Apple's Bluetooth low energy continuity protocol. *Proceedings on Privacy Enhancing Technologies* 2019, 4 (2019), 34–53. https://doi.org/10.2478/popets-2019-0057
[35] Martin Woolley, Bluetooth. 2020. *Bluetooth Core Specification Version 5.2 Feature Overview*. Technical Report. Bluetooth.
[36] Martin Woolley, Bluetooth. 2022. *The Bluetooth® Low Energy Primer Version 1.0.4*. Technical Report. Bluetooth.
[37] Travis Mayberry, Erik-Oliver Blass, and Ellis Fenske. 2023. Blind My—An Improved Cryptographic Protocol to Prevent Stalking in Apple's Find My Network. *Proceedings on Privacy Enhancing Technologies* 1 (2023), 85–97.
[38] Travis Mayberry, Ellis Fenske, Dane Brown, Jeremy Martin, Christine Fossaceca, Erik C Rye, Sam Teplov, and Lucas Foppe. 2021. Who Tracks the Trackers? Circumventing Apple's Anti-Tracking Alerts in the Find My Network. In *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society*. ACM, Korea, 181–186. https://doi.org/10.1145/3463676.3485616
[39] Michael Archambault. 2021. How to use Samsung Galaxy SmartTag technology.
[40] NutFind. 2013. Nut Find3 Smart Tracker. Retrieved Jan 2023 from https://www.nutfind.com/products/all-new-nut-find3-smart-tracker-never-lose-anything-4-pack-white-green-orange-grey
[41] Ron Amadeo. 2023. Google plans AirTag clone, will track devices with 3 billion Android phones.
[42] Thomas Roth, Fabian Freyer, Matthias Hollick, and Jiska Classen. 2022. AirTag of the Clones: Shenanigans with Liberated Item Finders. In *2022 IEEE Security and Privacy Workshops (SPW)*. IEEE, San Francisco, CA, 301–311. https://doi.org/10.1109/SPW54247.2022.9833881
[43] Samsung. 2021. Samsung Galaxy SmartTag. Retrieved Jan 2023 from https://www.samsung.com/us/mobile/mobile-accessories/phones/samsung-galaxy-smart-tag-1-pack-black-ei-t5300bbegus/
[44] Secure Mobile Networking. 2022. AirGuard - AirTag protection. Retrieved Jan 2023 from https://play.google.com/store/apps/details?id=de.seemoo.at_tracking_detection.release
[45] Charlie Sorrel. 2023. Tile's Odd New Anti-Theft Mode May Make Stalking Even Easier.
[46] Tile Inc. 2016. Tile Mate. Retrieved Jan 2023 from https://www.tile.com/products/tile-mate
[47] Todd Haselton. 2021. Here's how Apple's AirTag trackers compare to Tile, and why the company is so upset with Apple.
[48] Mira Weller, Jiska Classen, Fabian Ullrich, Denis Waßmann, and Erik Tews. 2020. Lost and found: stopping Bluetooth finders from leaking private information. In *WiSec '20: Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks)*. ACM, Austria, 184–194. https://doi.org/10.1145/3395351.3399422
[49] Tingfeng Yu, James Henderson, Alwen Tiu, and Thomas Haines. 2022. Privacy Analysis of Samsung's Crowd-Sourced Bluetooth Location Tracking System.

A SAFETY ALERTS

Figure 9 shows the safety alerts generated by the victim's device when three standard AirTags and one cloned AirTag (mimicking an AirPod with the MSN value of the Status byte in the BLE advertisement set to 3) consistently followed the victim. It is worth noting that during the experiments, all genuine AirTags consistently triggered safety alerts simultaneously on all devices, except for this specific case where there was a 2-minutes delay. This anomaly can be attributed to the presence of malformed packets and incorrect CRC packets in some of the BLE advertisements emitted by the ESP32 hardware utilized to prepare the cloned AirTag.



Tracking Notifications

AirPod Found Moving With You now
The owner of this item can see its location. Tap to open Find My and see available actions.

AirTag Found Moving With You 2m ago
The owner of this item can see its location. Tap to open Find My and see available actions.

AirTag Found Moving With You 2m ago
The owner of this item can see its location. Tap to open Find My and see available actions.

AirTag Found Moving With You 2m ago
The owner of this item can see its location. Tap to open Find My and see available actions.

Figure 9: Time-sensitive Safety Alert generated by the victim's device for 3 genuine AirTags and one stealthy cloned tag disguised as AirPod.

Table 7: Investigating the Impact of Phone Model on Safety Alerts. (s = Sample's standard deviation in minutes.)

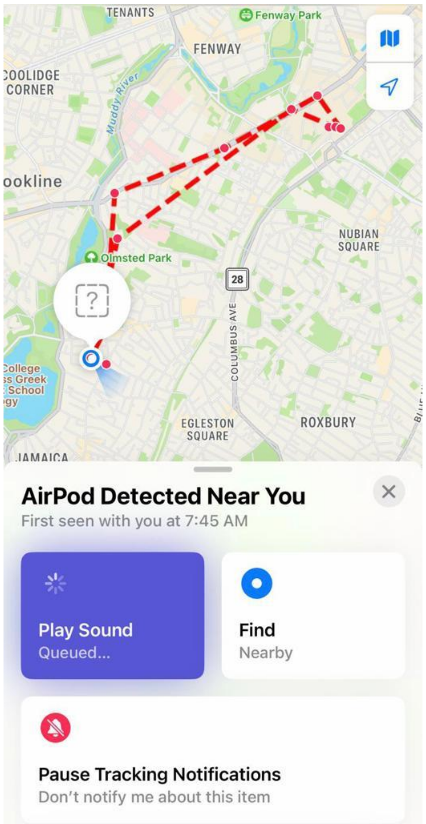
Day	Time after alert generated			s
	iPhone XR	iPhone 13 Pro	iPhone 13 mini	
1	8 hrs 21 mins	8 hrs 21 mins	8 hrs 22 mins	0.58
2	8 hrs 16 mins	8 hrs 15 mins	8 hrs 18 mins	1.53
3	8 hrs 36 mins	8 hrs 36 mins	8 hrs 35 mins	0.58

In addition, Figure 10 shows the route displayed to the victim when he opens up the safety alert for the AirPod. As ESP32 does not have a microphone, the victim can not play sound to locate the cloned AirTag, and thus we see that the play command is queued.

B EXPERIMENT RESULTS

This section presents the supporting results for some of the experiments conducted in Section 3. As such, we investigated the effect of various metrics on the generation of safety alerts, for instance, the victim's device model in Table 7, mode of transportation as driving in Table 8, accelerated alerts post-11 pm in Table 9, significant locations in Table 10, distance between AirTag and victim's device in Table 11 and people density in Table 12.

To quantify the differences in timings in minutes among victims' devices, we utilized the sample's standard deviation (s). Note that the results of Table 8 and Table 9 have been illustrated in Figure 4 and Figure 5. For other experiments, the variance between the alert times was very low (approximately 1 to 2 minutes) for experiments spanning over 8+ hours, thereby meaning that those metrics do not impact the generation of safety alerts.



AirPod Detected Near You
First seen with you at 7:45 AM

Play Sound
Queued...

Find
Nearby

Pause Tracking Notifications
Don't notify me about this item

Figure 10: Safety Alert for fake AirPod.

Table 8: Supporting Data for Figure 4 - Investigating the Impact of Mode of Transportation (Driving) on Safety Alerts. (Here, s = Sample's standard deviation in minutes.)

Start Time	Time after alert generated			s
	iPhone XR	iPhone 13 Pro	iPhone 13 mini	
5 am	8 hrs 21 mins	8 hrs 21 mins	8 hrs 22 mins	0.58
7 am	8 hrs 36 mins	8 hrs 36 mins	8 hrs 35 mins	0.58
9 am	8 hrs 16 mins	8 hrs 15 mins	8 hrs 18 mins	1.53
11 am	8 hrs 07 mins	8 hrs 06 mins	8 hrs 05 mins	1
1 pm	8 hrs 27 mins	8 hrs 14 mins	8 hrs 14 mins	7.5
3 pm	8 hrs 06 mins	8 hrs 05 mins	8 hrs 05 mins	0.58
5 pm	6 hrs 11 mins	6 hrs 11 mins	6 hrs 11 mins	0
7 pm	4 hrs 07 mins	4 hrs 03 mins	4 hrs 06 mins	2.1
9 pm	2 hrs 03 mins	2 hrs 03 mins	2 hrs 04 mins	0.58
11 pm	32 mins	31 mins	31 mins	0.58
1 am	36 mins	35 mins	36 mins	0.58
3 am	30 mins	32 mins	31 mins	1

C RETRIEVING ENCRYPTED LOCATION REPORTS

Figure 11 shows the encrypted location report retrieved through our modified OpenHaystack code for the public key of a legitimate AirTag. Although the adversary requires a private key to decrypt

Table 9: Supporting Data for Figure 5 - Investigating Accelerated Safety Alerts after 11 pm. The standard deviation of the sample (s) indicates the variation in results in minutes.

Start Time	Time when Safety Alert was generated			s
	iPhone XR	iPhone 13 Pro	iPhone 13 mini	
9:30 pm	11:02 pm	11:02 pm	11:03 pm	0.58
9:45 pm	11:03 pm	11:02 pm	11:03 pm	0.58
10:00 pm	11:04 pm	11:07 pm	11:06 pm	1.53
10:15 pm	11:03 pm	11:03 pm	11:04 pm	0.58
10:30 pm	11:07 pm	11:06 pm	11:07 pm	0.58
10:45 pm	11:18 pm	11:18 pm	11:21 pm	1.73
11:00 pm	11:35 pm	11:37 pm	11:37 pm	1.15

Table 10: Investigating the Impact of Visiting Significant Locations (SL) on Safety Alerts.

Start Time	Time after alert generated on reaching home		
	iPhone XR	**iPhone 13 Pro**	**iPhone 13 Mini**
	SL enabled	SL enabled	SL disabled
9 am	6 mins	6 mins	✗
11 am	2 mins	1 min	✗
1 pm	7 mins	4 mins	✗
3 pm	2 mins	2 mins	✗
5 pm	9 mins	7 mins	✗
7 pm	5 mins	4 mins	✗
11pm	3 mins	5 mins	✗

Table 11: Investigating Impact of Distance between AirTag and the Victim's Device on the Safety Alerts.

Day	Distance (m)						
	1	2	3	4	5	6	7
1	34 mins	34 mins	35 mins	37 mins			
2			33 mins	33 mins	✗	✗	
3				35 mins	✗	✗	✗

Table 12: Investigating the Impact of People Density on Safety Alerts.

Start Time	Safety Alert generated?		
	iPhone XR	**iPhone 13 Pro**	**iPhone 13 Mini**
Beach (before 8 hrs of exposure)			
7 am	✗	✗	✗
9 am	✗	✗	✗
Deserted street alongside Pond			
3 am	32 mins	31 mins	32 mins
3 am	28 mins	22 mins	28 mins
3 am	33 mins	33 mins	33 mins

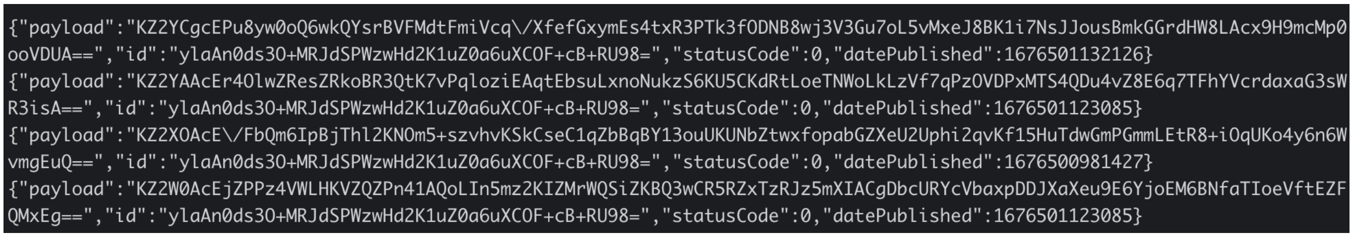
Table 13: Impact of Broadcast Intervals on Upload Timings: An Hour-Long Study of iPhones with the locked and unlocked screen.

Transmission Frequency (min)	Reports sent out	Reports received (Locked)	Reports received (Unlocked)
0.5	120	19	63
1	60	18	41
1.5	40	12	26
2	30	14	25
2.5	24	10	22
3	20	6	15
3.5	17	9	12
4	15	4	12
4.5	13	4	10
5	12	2	11
5.5	10	1	7
6	10	0	8
6.5	9	1	6

the report, Apple servers must not allow any user to fetch reports for anyone else. As recommended, this is possible if the server validates the mapping between the hash of the public key sent in the fetch request and the Apple ID requesting it.

D FINDER DEVICE'S REPORTING DYNAMICS

In our research, we aimed to determine the frequency at which a single iPhone reports the location of nearby AirTags to Apple's servers in an hour. To achieve this, we utilized OpenHaystack-based fake AirTags to control the number of advertisements sent by a genuine AirTag as a reference. Typically, an AirTag broadcasts BLE advertisements once every 2 seconds in disconnected mode. The experiment was conducted inside an RF shield (Faraday cage) to ensure that only one finder device uploads the location reports for subject tags. To allow the finder device to hear BLE advertisements and upload the location reports on the server, we set up a router inside the RF shield and enabled the device's Wi-Fi and Bluetooth. The results, shown in Table 13, indicate that a finder device does not upload location reports for every public key it listens to and uploads more reports when the finder device is turned on (i.e., the screen is not locked).



```
{"payload":"KZ2YCgcEPu8yw0oQ6wkQYsrBVFMdtFmiVcq\/XfefGxymEs4txR3PTk3fODNB8wj3V3Gu7oL5vMxeJ8BK1i7NsJJousBmkGGrdHW8LAcx9H9mcMp0
ooVDUA==","id":"ylaAn0ds3O+MRJdSPWzwHd2K1uZ0a6uXCOF+cB+RU98=","statusCode":0,"datePublished":1676501132126}
{"payload":"KZ2YAAcEr4OlwZResZRkoBR3QtK7vPqloziEAqtEbsuLxnoNukzS6KU5CKdRtLoeTNWoLkLzVf7qPzOVDPxMTS4QDu4vZ8E6q7TFhYVcrdaxaG3sW
R3isA==","id":"ylaAn0ds3O+MRJdSPWzwHd2K1uZ0a6uXCOF+cB+RU98=","statusCode":0,"datePublished":1676501123085}
{"payload":"KZ2XOAcE\/FbQm6IpBjThl2KNOm5+szvhvKSkCseC1qZbBqBY13ouUKUNbZtwxfopabGZXeU2Uphi2qvKf15HuTdwGmPGmmLEtR8+iOqUKo4y6n6W
vmgEuQ==","id":"ylaAn0ds3O+MRJdSPWzwHd2K1uZ0a6uXCOF+cB+RU98=","statusCode":0,"datePublished":1676500981427}
{"payload":"KZ2W0AcEjZPPz4VWLHKVZQZPn41AQoLIn5mz2KIZMrWQSiZKBQ3wCR5RZxTzRJz5mXIACgDbcURYcVbaxpDDJXaXeu9E6YjoEM6BNfaTIoeVftEZF
QMxEg==","id":"ylaAn0ds3O+MRJdSPWzwHd2K1uZ0a6uXCOF+cB+RU98=","statusCode":0,"datePublished":1676501123085}
```

Figure 11: Encrypted Location Report retrieved for an AirTag that belongs to another user