# AirCatch: Effectively tracing advanced tag-based trackers

Abhishek Kumar Mishra
abhishek.mishra@kit.edu
Karlsruhe Institute of Technology (KIT)

Swadeep
lnu.swa@northeastern.edu
Northeastern University

Guevara Noubir
g.noubir@northeastern.edu
Northeastern University

Mathieu Cunche
mathieu.cunche@inria.fr
University of Lyon / Inria

## Abstract

Tag-based tracking ecosystems help users locate lost items, but can also be leveraged for unwanted tracking and stalking. Existing protocol-driven defenses and prior solutions largely assume stable identifiers or predictable beaconing behavior. However, identifier-based defenses fundamentally break down against *advanced* rogue trackers that aggressively rotate identifiers and fragment transmissions to evade correlation. We present AirCatch, a passive detection system that exploits a fundamental physical-layer asymmetry: while logical identifiers can change arbitrarily fast, the transmitter's analog imprint remains stable and reappears as a *compact* and *persistently occupied* region in Carrier Frequency Offset (CFO) feature space. AirCatch advances the state of the art along three axes: (i) a *novel, modulation-aware* CFO fingerprint that augments packet-level CFO with transition-specific CFO components derived from BLE GFSK modulation behavior to amplify device distinctiveness; (ii) a *new tracking detection algorithm* based on density and persistence in CFO space that is robust to contamination and evasion through per-identifier segmentation; and (iii) an *ultra-low-cost* receiver, a ≈ $10 BLE SDR named *BlePhasyr*, built from commodity components, that makes RF-fingerprinting based detection practical. We evaluate AirCatch across multiple tag families in multi-hour captures, systematically stress-test advanced evasion techniques, and validate the system on diverse real-world mobility traces while varying background tag density. Across these stress tests, AirCatch achieves *zero observed false positives* and *early detection* across a broad range of adversarial configurations and environments, degrading gracefully only in extremely low-rate regimes that also reduce attacker utility.

## Keywords

Bluetooth Low Energy (BLE), RF fingerprinting, Carrier Frequency Offset (CFO), Unwanted tracking detection

## 1 Introduction

BLE-based item trackers (e.g., Apple AirTag, Samsung SmartTag, Google Find My Device tags, and Tile) have enabled crowd-sourced localization at a global scale, but have also been repeatedly repurposed for unwanted tracking and surveillance [3, 7, 19, 28, 34]. To mitigate abuse, ecosystems combine (i) protocol privacy mechanisms (rotating identifiers, cryptographic payloads) and (ii) platform/user-facing alerts intended to surface tags that move along with a victim for an extended period [11, 14, 29]. In practice, these defenses remain uneven and often do not work across ecosystems, and can be slow under real mobility and dense public settings [11, 18]. Moreover, they largely depend on *protocol-visible continuity*, MAC/payload correlation across time, which is exactly what an advanced rogue tracker can disrupt by rapidly rotating pseudonyms and duty-cycling transmissions (a capability already achievable with modified tags and open tooling) [15, 22]. As a result, stealthy attackers preserve tracking utility while depriving identifier-based detectors of persistent evidence.

We present AirCatch, a passive detector designed for this *advanced* regime. AirCatch exploits a physical-layer constraint: while logical identifiers can change arbitrarily fast, a transmitter's analog imprint is far harder to shed. In particular, the carrier frequency offset (CFO) induced by oscillator and synthesizer imperfections tends to reappear as a compact region in CFO space for a given device, even as MACs and payload fields churn. AirCatch turns this into an operational signature of unwanted tracking: an adversary that follows a victim induces a *persistently occupied, abnormally compact CFO core* that repeatedly accumulates many short-lived identifiers. To make this robust in heterogeneous, crowded traces, AirCatch (i) introduces a novel modulation-aware CFO fingerprint that augments packet-level CFO with bit-transition components to amplify distinctiveness, (ii) effectively uses per-ecosystem clustering in CFO space, and (iii) applies a persistence plus core-density based detection algorithm aligned with user-facing anti-tracking requirements.

Unlike prior CFO-enabled BLE defenses aimed at detecting spoofing/masquerade against stationary devices (e.g., BlueShield and BLEGuard, which modify commodity sniffers such as Ubertooth and rely on features like advertising interval, RSSI, and CFO) [6, 33], AirCatch targets *co-moving tracker detection under identifier rotation* across real mobility contexts and dense backgrounds, where transient encounters and ecosystem heterogeneity dominate. We also explicitly account for realistic background emitters observed in practice (e.g., AirPods-like behavior) when assessing robustness [25].

We evaluate AirCatch across various tag families with multi-hour captures, systematically stress-test evasion using a scenario generator over a grid of transmission periods (with per-transmission pseudonym rotation), and validate in diverse real mobility traces (home/office commutes, public transport, car travel) plus an airport stress test for false positives in a dense public environment. Across


*Proceedings on Privacy Enhancing Technologies YYYY(X), 1–18*

https://doi.org/XXXXXXX.XXXXXXX

these stress tests, AirCatch cleanly separates benign background traffic from advanced tracking behavior at conservative operating points, achieving zero observed false positives and reliable detection over a wide range of attacker configurations. It can potentially only degrade gracefully in extremely low-rate regimes that also reduce attacker utility. Our major contributions are fourfold:

**1. Modulation-aware RF fingerprints.** We introduce a modulation-aware physical-layer fingerprint that augments packet-level CFO with four transition-conditioned CFO components derived from BLE GFSK behavior. The resulting five-dimensional representation captures complementary transmitter-dependent structure that is averaged out by a prior scalar CFO estimate.

**2. Passive detection of advanced rogue trackers.** We present AirCatch, a passive anti-tracking detector that remains effective under *per-transmission identifier rotation* and *duty-cycled/stealthy* beaconing by leveraging a stable signal: a persistently compact and densely occupied CFO core.

**3. Low-cost SDR-based BLE anti-tracking device.** We design and release *BlePhasyr*, a pocket-size, cheapest-to-date (≈$10) BLE *micro-SDR* that provides CFO-grade physical-layer evidence for passive monitoring, making anti-tracking practical for everyday, resource-constrained deployments.

**4. End-to-end system and real-world validation.** We provide a companion Android application that surfaces actionable user alerts and optional technical insights, and we demonstrate robust AirCatch performance via an extensive evaluation spanning multiple tracker ecosystems, adversarial configurations, and real-world mobility environments.

## 2 Related Works

*Commodity BLE tracker ecosystems and protocol behavior.* A growing body of work reverse-engineers and characterizes crowd-sourced finding ecosystems, including Apple AirTag [21, 28], Samsung SmartTag [34], Google Find My Device [3], and Tile [19]. Beyond protocol structure, recent studies analyze real-world behavior and user-facing safety properties such as alert latency, delivery delays, and cross-ecosystem coverage [11, 16, 18, 32]. Prior work also highlights additional ways these ecosystems get repurposed (e.g., leveraging tags for localization) [13] or turning commodity devices into emitters compatible with offline-finding infrastructures [8]. We complement these efforts with a consolidated protocol-level analysis of identifier rotation and safety-relevant state behavior across ecosystems (§A), distilling the privacy and linkability implications that motivate our threat model. While these efforts clarify the operational landscape and design trade-offs, they do not provide a solution that remains reliable when a tracker *actively* manipulates identifier exposure (fast MAC rotation and stealthy transmission) across heterogeneous environments.

*Identifier-layer fingerprinting, linkability, and evasion.* At the MAC layer, BLE includes standardized address randomization, and empirical studies confirm that major mobile platforms implement it effectively [2]. However, linkability can persist through ecosystem-specific design choices and unsynchronized rotations between MAC and payload fields; for instance, prior work (and our observations) reports long-lived identifiers in some tracker families (notably Tile) [19, 27]. More fundamentally, custom emitters can directly manipulate advertisement identifiers and rotation schedules, undermining any detector that assumes continuity in protocol-visible fields [15]. As a result, identifier-layer correlation alone is not reliable for anti-tracking against an *active* adversary.

*Anti-stalking defenses.* Platform defenses and third-party tools attempt to surface co-moving trackers across ecosystems, including cross-ecosystem detection on Android for Apple Find My devices [14] and analyses of Apple's safety alerts and limitations [29]. Additional systems propose improvements to detection and localization workflows or measurement of network reporting behavior [4, 9, 17, 23, 24]. Broader surveys of covert tracking and intimate partner surveillance emphasize that commodity surveillance devices remain readily accessible and that detection tooling is uneven in practice [7]. Recent work also explores algorithmic directions for improving anti-abuse detection pipelines [10, 22]. Nonetheless, these approaches predominantly rely on identifier continuity and/or ecosystem cooperation, and thus offer limited guarantees of fast pseudonym rotation and rate manipulation.

*Protocol analyses and physical-layer fingerprinting.* Security analyses of BLE proximity tracking protocols identify design choices and attack surfaces that influence both privacy and detectability [20]. Separately, physical-layer fingerprinting leverages hardware impairments such as CFO, IQ imbalance, and transients [5]. In BLE, physical-layer signatures (including CFO) have been shown to enable device tracking [12], and subsequent work explores CFO-derived features for device discrimination or tracking [30, 31]. Defenses that obfuscate physical-layer fingerprints (including CFO manipulation) have also been explored for certain chipsets [26]; however, these proposals rely on experimental BLE-compatible transmitter chipsets with explicit support for CFO manipulation, a capability not exposed by commodity BLE chipsets, particularly those used in current commercial trackers. In essence, prior work on physical-layer fingerprinting does not address the core anti-stalking requirement we target: a *deployable, low-cost* solution that produces evidence of *persistent* tracking behavior under adversarial identifier churn.

*Summary.* Prior work (i) characterizes tracker ecosystems and their safety mechanisms, (ii) evaluates and improves identifier-driven detectors, (iii) demonstrates that motivated adversaries can disrupt identifier-layer continuity, and (iv) shows that physical-layer signatures can support BLE device fingerprinting. Table 1 summarizes how these approaches differ from AirCatch across threat model, identifier dependence, adversarial robustness, deployment requirements, ecosystem scope, and detection delay. In particular, AirCatch goes beyond standard packet-level CFO analysis by introducing novel transition-conditioned CFO features. This enables detection under per-transmission identifier rotation and adaptive transmission schedules using a low-cost receiver.

## 3 Threat Model and Adversary

We study the detection of *rogue BLE trackers* that remain co-located with a victim over time while *actively evading protocol-layer linkability*. The defender is a small commodity passive sniffer that

| Approach | Goal/Threat model | ID dependency | Robustness | Deployment | Scope | Delay | Remark |
|---|---|---|---|---|---|---|---|
| ***Physical-layer methods*** | | | | | | | |
| CFO Tracking [12] | BLE device fingerprinting under MAC randomization | No | N/A | SDR; > $300 | General BLE | N/A | Links transmissions using RF fingerprints |
| BlueShield [33] | BLE spoofing detection | N/A | N/A | Ubertooth; ≈ $100 | General BLE | N/A | Detects impersonation of known devices |
| ***Platform-native protections*** | | | | | | | |
| Apple alerts [29] | Study on Lost Mode Tracker | Yes | No | Native iOS | Apple Airtags | 8h in the Morning | Long alert delays reported |
| Google alerts [3] | Study on Lost Mode Tracker | Yes | No | Native Android | Google FMDN | Not Reported | Implementation limitations reported |
| ***Third-party tools*** | | | | | | | |
| AirGuard [14] | Cross-ecosystem tracker detection | Yes | No | Smartphone app | Major ecosystems | ≈ 30 min | Broader coverage than native alerts |
| BLE-Doubt [4] | BLE co-travel detection | Yes | No | App + GPS | General BLE | ≈ 10 min | Uses distance and duration thresholds |
| ***Other proposals*** | | | | | | | |
| DeTagTive [9] | Link scheduled address rotations | Partial | No | Raspberry Pi | Ecosystem-agnostic | 25–30 min | Detect Rotating (15 mins) MAC address using RSSI. |
| Blind My [22] | Prevent multi-tag emulation | Yes | N/A | Protocol change | Apple Find My | N/A | Cryptographic protocol proposal |
| Abuse-Resistant OF [10] | Prevent offline-finding abuse | No | No | Protocol/backend change | Vendor adoption | 39-47 min | Abuse-resistant protocol proposal |
| **AirCatch** | **Detect fast identifier-rotating, rate-adaptive trackers** | **No** | **Yes** | ***BlePhasyr*, app; ≈ $10** | **Apple, Google, Samsung, Tile, custom** | **40 min** | **Detects persistent emitters across short-lived IDs** |

**Table 1: Comparison with prior physical-layer and anti-tracking approaches. ID dependency: requires protocol-visible identity continuity. Robustness: Detection against rogue tags.**

observes BLE advertisements and extracts timestamps $t_k$, advertiser addresses $m_k$, payloads $p_k$, and CFO-derived feature vectors $\mathbf{f}_k$ (Section 4). The detector flags *persistently occupied* and *abnormally compact* regions in CFO-feature space that aggregate evidence across many short-lived identifiers (Section 5).

*Goals.* The primary goal is tracking safety: detecting a nearby device that persistently follows the victim, even under identifier churn. Operationally, we target near-zero false alarms in crowded environments. We do not aim to perfectly separate *all* surrounding devices or attribute a flagged device to a specific owner; rather, we detect the signature of tracking as joint CFO-space compactness and persistence.

*Defender assumptions.* We assume the adversary's transmissions are observable with non-trivial probability when the tracker is near the victim (a requirement for tracking functionality). We do not assume calibrated receivers or access to cloud identifiers, pairing secrets, or stable BLE addresses; our evaluation spans heterogeneous receivers, and our pipeline uses within-type normalization and robust statistics to mitigate receiver bias and transient outliers.

*Tracker assumptions.* We focus on advertisements emitted in the ecosystem-specific state used when a tracker is separated from its owner and participates in crowdsourced localization and unwanted-tracking detection. We refer to these states collectively as *Lost Mode*, although their precise semantics differ across ecosystems: Apple distinguishes owner-near and separated operation, Samsung defines connected, premature-lost, lost, and overmature-lost states, and Google defines an Unwanted Tracking state. Appendix A describes these state machines and their corresponding identifier-rotation behavior. Our modified trackers preserve the advertisement structure associated with the relevant Lost Mode state while manipulating their identifiers and transmission schedules. `AirCatch` filters benign persistent companions based on advertisement state and their characteristically low identifier churn, while treating rapid identifier churn as indicative of the adversarial behavior. The former pattern, trackers following the ecosystem-defined rotation schedules, is already covered by existing unwanted-tracking detection, so `AirCatch` need not treat it specially.

## 3.1 Adversary Model

*Objective.* The adversary's goal is to track the victim by keeping a device co-located for an extended time while avoiding detection by rotating identifiers, reducing linkability, and shaping emissions.

*Capabilities.* We assume firmware-level control typical of a determined attacker: (i) arbitrary advertiser-address rotation and payload-field changes, (ii) duty-cycling and rate shaping over a wide range (seconds to minutes), (iii) emission-pattern shaping to mimic benign traffic, and (iv) operation under mobility and interference (home/office, public transport, car). We assume the adversary knows the detection algorithm and thresholds and may tune behavior accordingly.

*Physical and functional constraints.* Despite protocol control, the adversary faces two constraints exploited by our design. First, for a fixed transmitter, oscillator, and analog impairments induce a CFO signature that is stable relative to identifier churn (Section 4); rotating identifiers does not change the underlying hardware imprint. Second, meaningful tracking requires *persistence*: the tracker must reappear across time and contexts. Extended periods of silence reduce tracking utility; continued functionality forces repeated emissions that accumulate evidence across windows.

*Threats addressed.* Our detector is designed to withstand: (i) *fast identifier rotation*: segmentation prevents cross-device contamination and rotation increases the number of segments contributed by the same physical emitter, strengthening persistence/density

evidence; (ii) *duty-cycling*: we quantify and tune robustness in low-support regimes via windowing and robust embedding features; and (iii) *crowded environments and mobility variation*: within-type clustering and persistence thresholds reject transient passers-by even when locally dense.

*Adversary instantiations.* We use two complementary adversary constructions. First, our *functional ESP32 adversary* runs modified OpenHaystack firmware. It can change its advertiser address at every transmission and independently control its advertising period. Second, our *Apple and Google replay adversaries* are trace-based constructions derived from captures of commercial tags. We retain the measured physical-layer features of the commercial transmitter, then downsample its emissions to the target advertising period and assign a new logical pseudonym to every retained transmission. These replay adversaries let us evaluate `AirCatch` against physical-layer distributions from commercial hardware. `AirCatch` separates concurrent trackers by their distinct CFO regions; although an adversary could dilute persistence by rotating across a larger tracker pool, doing so substantially increases hardware, placement, synchronization, and maintenance costs.

## 3.2 Limitations and Out-of-Scope Attacks

*Hardware-level CFO obfuscation.* Deliberate transmitter-side CFO perturbation is outside `AirCatch`'s current threat model. An adaptive adversary with fine-grained control over the transmitter's physical layer could randomize or compensate CFO across identifiers, fragmenting the CFO cluster and evading the grouping. Such control is not typically exposed by commodity BLE chipsets, particularly those used in commercial tracking tags, and existing work in this direction relies on a custom BLE-compatible transmitter prototype [26]. We therefore treat this as a strictly stronger attacker class and do not claim robustness against it: our contribution is to show that victim-side detection is already practical with commodity trackers and low-cost sensing hardware, using a feature pipeline that extends beyond CFO toward richer RF fingerprints. Evaluating such countermeasures and possible defenses is left for future work.

*Near-total silence and receiver manipulation.* If a tracker transmits so rarely that it falls below practical capture probability, any passive detector will fail; such an operation also limits tracking utility, and our results quantify this trade-off. Receiver jamming and physical compromise are beyond the scope of the investigation.

# 4 Designing Effective Fingerprints

This section describes the fingerprint design used by `AirCatch`. Our goal is to derive *device-distinct* and *time-stable* features from passively observed BLE packets, while remaining robust to aggressive identifier rotation. We focus on CFO as a physically grounded proxy for oscillator-level imperfections, and we further decompose CFO by *symbol transition type* to expose additional hardware-dependent structure.

## 4.1 Challenges of State-of-the-Art Solutions

Prior work on radiometric fingerprinting has demonstrated that RF front-end imperfections (e.g., oscillator offset, IQ imbalance, phase noise, PA non-linearities) can form device-distinct signatures under passive observation [5]. However, applying these techniques to commodity BLE trackers raises several practical challenges:

*Protocol-layer churn and short packets.* Modern tracking ecosystems rotate identifiers and may randomize advertisement payloads, making fingerprints based on stable identifiers, long preambles, or repeated known sequences brittle when the observable structure is short-lived or partially unknown.

*Channel confounding.* Many fingerprints rely on amplitude- or spectrum-based features (e.g., PSD shape, transient energy, rise/fall times). These can vary significantly with multipath, relative motion, antenna orientation, and receiver front-end filtering, confounding device effects with environment effects. Robust designs should minimize sensitivity to unknown channel gain and phase.

*Receiver dependence and sampling constraints.* Fine-grained transient/spectral approaches often require high-rate IQ captures, consistent receiver calibration, and strict alignment across packets. In contrast, our setting targets practical deployments: commodity receivers, varying capture quality, and heterogeneous packet contents.

*Adversarial adaptation.* Recent work explicitly identifies the CFO as a privacy-relevant fingerprint in BLE, indicating both its strength and its adversarial relevance [26]. A robust fingerprint must exploit CFO while also extracting additional structure beyond a single scalar offset.

## 4.2 CFO as a Proxy

CFO mainly stems from transmitter–receiver oscillator mismatch and appears as a constant residual rotation in the received complex baseband. Let the received samples be

$$r[n] \;=\; s[n]\; e^{j(2\pi\Delta f\, n/f_s+\phi_0)} \;+\; w[n], \tag{1}$$

where $s[n]$ is the transmitted baseband waveform, $\Delta f$ is the carrier-frequency offset, $f_s$ is the sampling rate, $\phi_0$ is the initial carrier phase offset between the transmitter and receiver oscillators, and $w[n]$ is the additive noise. Under this model, the one-step product

$$z[n] \;=\; r[n]\; r^*[n-1] \tag{2}$$

has expected phase $\arg(z[n]) \approx 2\pi\Delta f/f_s$ plus modulation/noise terms. Summing these phasors across a packet suppresses zero-mean modulation contributions and yields a low-complexity CFO estimate:

$$\widehat{\Delta f}_{\text{packet}} \;=\; \frac{f_s}{2\pi}\; \arg\Big(\sum_{n=1}^{N-1} r[n]\; r^*[n-1]\Big). \tag{3}$$

This estimator is attractive for BLE-scale packets: it is $O(N)$, requires no pilots, and is compatible with gated windows (e.g., restricting to structurally reliable sub-regions of a packet). CFO has been used as a stable physical-layer identifier and as a discriminative signal statistic in multiple wireless contexts, including BLE [1, 31].

We estimate CFO over the exact complex baseband interval corresponding to a successfully decoded BLE packet, using decoder-provided start and end sample indices to isolate the packet waveform in IQ space. Optional amplitude-based gating suppresses residual preamble leakage and out-of-packet samples prior to estimation. Given the gated IQ sequence $x = \{x[n]\}_{n=0}^{N-1}$, we compute the packet-level CFO estimate $\widehat{\Delta f}_{\text{packet}}$ using Eq. 3.

### 4.3 Transition-Type CFOs as Distinct Features

A key limitation of a single CFO scalar is that it primarily captures the oscillator mismatch $\Delta f$, which may be similar across devices from the same vendor/batch. To extract additional device-dependent structure, we condition CFO estimation on *bit transition types*. Transition-conditioned features can significantly improve identifiability by capturing imperfections that manifest during symbol transitions.

In Gaussian Frequency Shift Keying (GFSK), the instantaneous frequency is shaped by a Gaussian filter whose impulse response spans multiple symbol intervals. As a result, the discriminator output at time $n$ reflects not only the current bit $b_i$, but also residual phase and frequency contributions from preceding symbols. For *equal transitions* ($0 \to 0$ and $1 \to 1$), the frequency trajectory remains on the same side of the modulation center, and the filter state reinforces a steady frequency bias dominated by the transmitter's oscillator offset and static analog impairments.

In contrast, *jump transitions* ($0 \to 1$ and $1 \to 0$) force a rapid sign change in the frequency deviation, causing transient asymmetries in the discriminator response due to finite filter memory, PLL settling dynamics, and DAC/PA slew limitations. These asymmetries are hardware-dependent and persist across packets, even when protocol-layer identifiers and payloads change. By estimating CFO separately over equal and jump transitions using the same phasor-sum estimator, we isolate these transition-dependent biases, yielding features that are both physically grounded and complementary to packet-level CFO.

*Bit-aligned partitioning.* Let $b_i \in \{0, 1\}$ denote the recovered BLE bit sequence for a decoded packet, excluding the preamble, and let each bit occupy $sps$ complex baseband samples. We partition the IQ stream into symbol-aligned windows

$$\mathcal{W}_i = \{\, n \mid n \in [i \cdot sps, (i+1) \cdot sps) \,\}, \tag{4}$$

where $\mathcal{W}_i$ contains the samples associated with bit $b_i$. For each adjacent bit transition ($b_{i-1} \to b_i$), we assign the corresponding differential phasor products $x[n]x^*[n-1]$, including the boundary product at the start of $\mathcal{W}_i$, to one of four transition-conditioned accumulators

$$T_{00},\ T_{11},\ T_{10},\ T_{01}, \tag{5}$$

corresponding to transitions $0 \to 0$, $1 \to 1$, $1 \to 0$, and $0 \to 1$, respectively. For each transition class $uv \in \{00, 11, 10, 01\}$, we compute a class-specific CFO estimate using the same differential phase estimator as Eq. 3, restricted to the phasor subset associated with that transition type:

$$\widehat{\Delta f}_{uv} \;=\; \frac{f_s}{2\pi}\ \arg\Big(\sum_{(n \in T_{uv})} x[n]\ x^*[n-1]\Big). \tag{6}$$

This construction captures transition-conditioned frequency behavior induced by BLE GFSK modulation while preserving estimator consistency. The union of all class-specific phasor sets exactly reconstructs the global packet-level phasor sum:

$$\sum_{uv} \sum_{n \in T_{uv}} x[n]x^*[n-1] \;=\; \sum_{n=1}^{N-1} x[n]x^*[n-1], \tag{7}$$

ensuring that recombination recovers the original packet-level estimate $\widehat{\Delta f}_{\text{packet}}$ computed over the same IQ interval. This property imposes an internal consistency constraint and prevents potential estimator drift caused by partition misalignment.

### 4.4 Holistic Fingerprints

We construct a *holistic* RF fingerprint that jointly captures coarse oscillator mismatch and transition-conditioned modulation behavior from the same decoded BLE packet. Given a gated packet waveform, we extract the following five-dimensional fingerprint vector:

$$\mathbf{f}_{\text{IQ}} = \big[\widehat{\Delta f}_{\text{packet}},\ \widehat{\Delta f}_{00},\ \widehat{\Delta f}_{11},\ \widehat{\Delta f}_{10},\ \widehat{\Delta f}_{01}\big], \tag{8}$$

where $\widehat{\Delta f}_{\text{packet}}$ denotes the packet-level CFO estimate, and $\widehat{\Delta f}_{uv}$ denotes the transition-conditioned CFO associated with bit transition $u \to v$.

In settings where full-rate complex IQ is not available (e.g., Ubertooth-class devices), the receiver exposes only a coarse per-byte frequency proxy rather than continuous complex samples. For this case, we engineer an alternative *content-stratified* CFO fingerprint based on Hamming-weight conditioning of the proxy; we describe the construction, rationale, and high effectiveness in Appendix §B. Intuitively, this stratification recovers discriminative structure that is otherwise averaged out by a single coarse CFO mean, providing a practical fallback when only limited radio statistics are observable. In this paper, our primary focus is on *BlePhasyr*, our ultra-low-cost BLE *micro-SDR* that provides full I/Q for robust CFO and transition-CFO extraction.

*Novelty.* Our holistic fingerprint departs from prior CFO-only approaches in two fundamental ways. First, we introduce a *modulation-aware decomposition* of CFO via explicit bit transitions, grounded in the same physical origin: transmitter-specific modulation dynamics shaped by Gaussian filtering and analog front-end imperfections. Second, we treat these conditioned CFOs as *first-class radiometric features*, yielding a compact and interpretable 5-dimensional representation that is robust to protocol-layer identifier rotation and independent of payload semantics. This directly targets the operational regime of BLE trackers, where identifiers and payloads evolve rapidly, but hardware-induced impairments persist across time and packets [5, 26].

## 5 Tracking Detection

Our detection pipeline targets evasive BLE trackers that aggressively rotate identifiers and vary transmission behavior to avoid logical-layer correlation. Although identifiers and beaconing patterns can change arbitrarily fast, the transmitter's analog RF characteristics remain comparatively stable over time. Consequently, packets emitted by the same physical device induce recurring structure in CFO feature space, forming a persistently re-occupied and abnormally compact geometric region across temporal windows. AirCatch detects trackers by identifying these dense and temporally persistent CFO-space regions while aggregating evidence across many short-lived logical identifiers and noisy mobility environments.

**Notation.** Each decoded packet $k$ is associated with timestamp $t_k \in \mathbb{R}_{\geq 0}$, advertiser address $m_k \in \mathcal{M}$, payload $p_k$, and CFO feature vector $\mathbf{f}_k \in \mathbb{R}^{d_0}$, where $d_0 = 5$ corresponds to the holistic CFO representation defined in Eq. 8. We partition time into fixed windows of duration $W > 0$ and define the corresponding window index

**Algorithm 1:** `AirCatch` Tracker Detection

**Input:** Packet stream $(t, m, p, \mathbf{f})$; window $W$; block size $B$; thresholds $(\delta, T_{\min}, T_{\text{gap}})$.
**Output:** Adversary alerts $\mathcal{A}_j$.
Initialize persistence state $\Pi$ and packet buffer $\mathcal{P}$;
**foreach** *block $j$ of duration $B$* **do**
  Collect packets $(t, m, p, \mathbf{f})$ into $\mathcal{P}$;
  **(1) Segmentation:** Partition packets into segments $s = (u, \text{eco}(p), \text{dev}(p, m))$ and compute segment-level CFO centroids $\bar{\mathbf{f}}_s$;
  **(2) Type-wise clustering:** Cluster segments independently per ecosystem using standardized CFO embeddings;
  **(3) Core density estimation:** For each cluster $\tau$, compute a CFO-space core around the medoid and estimate $\text{coreDens}_\tau$;
  **(4) Persistent tracker detection:** Mark block $j$ positive if any cluster satisfies $\text{coreDens}_\tau \geq \delta$;
  **if** *block $j$ is positive* **then**
    Update persistence episode $\Pi$ using positive cluster timestamps;
    **if** *episode duration exceeds $T_{\min}$* **then**
      Emit adversary alert $\mathcal{A}_j \leftarrow \{\text{ADV}\}$ and reset $\Pi$;
  **else if** *time since last positive exceeds $T_{gap}$* **then**
    Reset persistence episode $\Pi$;

as $u_k \triangleq \lfloor t_k / W \rfloor$. Let $\text{eco}(\cdot)$ denote a coarse ecosystem classifier mapping payloads to {Apple, Google, Tile, Samsung, Unknown}. We further define a segmentation identifier $d_k = \text{dev}(p_k, m_k)$ corresponding to the logical device identity exposed by the protocol. For non-Samsung ecosystems, $d_k$ corresponds directly to the advertiser address $m_k$, whereas for Samsung trackers, we derive $d_k$ from a payload-embedded private identifier (PRIVID).

## 5.1 Step 1: Segmentation

Naive aggregation of packets across crowded temporal windows mixes unrelated emitters, smoothing CFO structure, and destroying the compactness and persistence properties required for robust tracking detection. We therefore perform segmentation at the granularity

$$s \triangleq (u_k,\ \text{eco}(p_k),\ d_k), \tag{9}$$

such that each segment contains packets associated with a single device identifier, ecosystem type, and temporal window ($W = 120$s). This segmentation strategy is resilient to adversarial transmission behavior: low-duty-cycle transmissions remain isolated rather than diluted within pooled aggregates, while aggressive identifier rotation increases the number of temporally consistent segments originating from the same physical transmitter, thereby strengthening persistence evidence in feature space. For each segment $s$ with associated packet index set $\mathcal{K}(s)$, we compute the segment-level feature centroid

$$\bar{\mathbf{f}}_s \triangleq \frac{1}{|\mathcal{K}(s)|} \sum_{k \in \mathcal{K}(s)} \mathbf{f}_k, \qquad n_s \triangleq |\mathcal{K}(s)|, \qquad (t_s^{\min}, t_s^{\max}), \tag{10}$$

where $n_s$ denotes the packet count and $(t_s^{\min}, t_s^{\max})$ captures the temporal extent of the segment.

## 5.2 Step 2: Type-wise embedding and clustering

To prevent cross-ecosystem mixing effects, we cluster segments independently within each ecosystem class. For each segment $s$, we first standardize the segment-level CFO centroid using type-specific statistics:

$$\mathbf{y}_s \triangleq \text{ZSCORE}(\bar{\mathbf{f}}_s). \tag{11}$$

We then augment the normalized representation with robust deviation features computed relative to the ecosystem-level median CFO vector $\tilde{\mathbf{c}}$:

$$d_s^{(2)} \triangleq \|\bar{\mathbf{f}}_s - \tilde{\mathbf{c}}\|_2, \qquad d_s^{(1)} \triangleq \|\bar{\mathbf{f}}_s - \tilde{\mathbf{c}}\|_1. \tag{12}$$

The resulting embedding $\mathbf{x}_s$ captures both normalized local structure and absolute deviation geometry in CFO space. We select the number of clusters $K$ using silhouette analysis over a bounded candidate grid and apply Ward-linkage agglomerative clustering to $\{\mathbf{x}_s\}$, yielding CFO-consistent regions $\tau$ that preserve persistent physical-layer structure without requiring perfect per-device separation.

## 5.3 Step 3: Core selection

For each cluster $\tau$, we perform compactness analysis exclusively in the standardized CFO space $\mathbf{y}_s$, rather than the augmented embedding $\mathbf{x}_s$, so that density estimates reflect intrinsic analog RF geometry rather than auxiliary deviation features. Let $\mathcal{S}_\tau$ denote the set of segments assigned to cluster $\tau$ and $n_\tau = |\mathcal{S}_\tau|$. We first compute the CFO-space medoid

$$s^\star \triangleq \arg\min_{s \in \mathcal{S}_\tau} \sum_{s' \in \mathcal{S}_\tau} \|\mathbf{y}_s - \mathbf{y}_{s'}\|_2, \tag{13}$$

which serves as the robust geometric center of the cluster. For each segment $s \in \mathcal{S}_\tau$, we then compute the radial distance

$$d_s \triangleq \|\mathbf{y}_s - \mathbf{y}_{s^\star}\|_2. \tag{14}$$

To isolate the stable high-density region associated with a persistent physical emitter, we define a dispersion-controlled core using the empirical distance distribution:

$$\mu_\tau \triangleq \frac{1}{n_\tau} \sum_{s \in \mathcal{S}_\tau} d_s, \qquad \sigma_\tau \triangleq \sqrt{\frac{1}{n_\tau} \sum_{s \in \mathcal{S}_\tau} (d_s - \mu_\tau)^2}, \tag{15}$$

$$\text{Core}_\lambda(\tau) \triangleq \{\, s \in \mathcal{S}_\tau :\ d_s \leq \mu_\tau + \lambda \sigma_\tau \,\}, \tag{16}$$

where $\lambda$ controls the admissible dispersion radius. We then estimate the effective core radius from the restricted core distribution:

$$\tilde{r}_\tau^{\text{core}} \triangleq \max\Big(\mu_\tau^{\text{core}} + \lambda \sigma_\tau^{\text{core}},\ r_{\min}\Big), \tag{17}$$

where $\mu_\tau^{\text{core}}$ and $\sigma_\tau^{\text{core}}$ denote the mean and standard deviation of $\{d_s\}$ computed only over $\text{Core}_\lambda(\tau)$. The lower bound $r_{\min}$ prevents unstable density inflation in near-degenerate clusters.

Finally, we quantify identifier diversity and normalized core density as

$$\text{macDiv}_\tau \triangleq \frac{M_\tau}{n_\tau}, \qquad \text{coreDens}_\tau \triangleq \frac{\text{macDiv}_\tau}{\tilde{r}_\tau^{\text{core}} + \epsilon}, \tag{18}$$

where $M_\tau$ denotes the number of distinct logical identifiers observed within cluster $\tau$. In our implementation, we use $\lambda = 1.5$ and $r_{\min} = 0.15$.

### 5.4 Step 4: Persistent tracker detection

We process BLE observations as a temporal stream partitioned into fixed-duration analysis blocks of length $B$ (default $B$=2400s). Steps 1–3 are executed independently within each block, producing a set of clusters and their associated compactness statistics. For block $j$, we first derive a block-level positive indicator based on the existence of at least one sufficiently dense CFO-consistent region:

$$\text{pos}_j \triangleq \exists\, \tau \text{ s.t. } \text{coreDens}_{\tau,j} \geq \delta. \tag{19}$$

This criterion captures the presence of a compact and identifier-diverse physical-layer structure consistent with a persistent transmitter undergoing identifier rotation.

To distinguish sustained tracking behavior from transient environmental encounters, we aggregate positive evidence across time using a persistence episode state. For every positive block ($\text{pos}_j$ = True), we compute the temporal extent covered by all positive clusters:

$$t^{\min}_{+,j} \triangleq \min_{\tau:\ \text{coreDens}_{\tau,j}\geq\delta} t^{\min}_{\tau,j}, \qquad t^{\max}_{+,j} \triangleq \max_{\tau:\ \text{coreDens}_{\tau,j}\geq\delta} t^{\max}_{\tau,j}, \tag{20}$$

where $t^{\min}_{\tau,j}$ and $t^{\max}_{\tau,j}$ are induced from the segment-level temporal bounds associated with cluster $\tau$. The persistence episode is then updated by maintaining the earliest observed positive time: $t^{\text{start}} \leftarrow \min(t^{\text{start}}, t^{\min}_{+,j})$, the latest observed positive time: $t^{\text{end}} \leftarrow \max(t^{\text{end}}, t^{\max}_{+,j})$, and the timestamp of the most recent positive observation: $t^{\text{last+}} \leftarrow t^{\max}_{+,j}$.

At the end of each block, we declare a persistent tracking adversary whenever the accumulated episode duration exceeds a minimum persistence threshold:

$$t^{\text{end}} - t^{\text{start}} \;\geq\; T_{\min}. \tag{21}$$

This persistence criterion suppresses short-lived encounters and focuses detection on sustained co-mobility patterns indicative of stalking behavior. To forget transient false positives and stale evidence, we reset the episode whenever a sufficiently long inactivity gap is observed after a negative block:

$$t^{\text{end}}_j - t^{\text{last+}} \;>\; T_{\text{gap}}, \tag{22}$$

where $t^{\text{end}}_j$ denotes the maximum packet timestamp observed in the current block. In our implementation, we use $T_{\text{gap}}$ = 1 day. Once an adversary is flagged, the episode state is reset to prevent repeated alerts for the same persistent tracker.

## 6 Anti-Tracking Device: *BlePhasyr*

AirCatch is implemented as a pocket-size, ≈*$10* SDR-based BLE *micro-SDR* named *BlePhasyr*. Our prototype uses a commodity *Seeed XIAO MG24* (Silicon Labs EFR32xG24)[1] as the RF front-end and a low-cost *ESP32-S3* companion module[2] (no other parts are required). The MG24 exposes raw receive IQ via the RAIL radio interface and streams it to a host over high-speed SPI. Figure 1 shows the complete pipeline and end-to-end UX: *BlePhasyr* continuously monitors BLE advertising channels and collects *IQ Samples at 1.34 MSPS*, buffers samples via DMA into a ring, and *SPI-streams* triggered IQ bursts to a host where we run BLE decoding and CFO estimation. In our implementation, the MG24 streams IQ over SPI to a small companion ESP32-S3 (used as a simple, high-throughput transport), which then relays the data to the phone over a wired USB-C link.

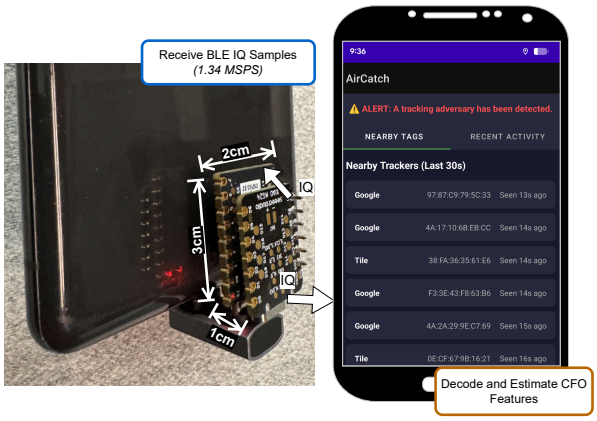



**Figure 1: *BlePhasyr*. The BLE micro-SDR receives IQ data and sends triggered IQ bursts to a host for BLE decoding and CFO estimation. The Android app presents an end-user alert and optional technical views of nearby tag activity.**

The companion Android application provides a user-facing safety interface: it surfaces a clear *tracking alert* when AirCatch flags an adversary, while also offering an "inspect" view that lists nearby tag activity (e.g., per-ecosystem sightings, recency, and identifiers when visible) to support transparency and debugging without requiring any specialized RF expertise.

On the firmware side, *BlePhasyr* solves the main physical bottleneck that makes "RF-fingerprinting-grade" sensing difficult on ultra-low-cost devices: sustaining high-throughput IQ capture without overflowing the Rx FIFO or stalling the radio. Concretely, the implementation uses (i) a large multi-chunk ring buffer (≈240 KB of staging) to absorb bursty BLE activity, (ii) tight critical-section queueing to maintain lock-free producer/consumer behavior between the Rx FIFO handler and the SPI transport, and (iii) a *polling* LDMA-driven SPI transmit path that minimizes IRQ overhead while sustaining a multi-Mbps stream (13 MHz EUSART/SPI clock).

To avoid streaming noise-dominated, low-information samples and to maintain practical power and throughput, *BlePhasyr* employs an energy-triggered capture state machine with a short pre/post history window: it computes a lightweight IQ energy statistic on each chunk and only transmits chunks that exceed a self-tuned threshold, with bounded pre/post capture to retain synchronization context. The firmware further hardens continuous operation with explicit overflow recovery and per-chunk metadata stamping (seed + checksums) to enable end-to-end integrity checks on the host. Together, these engineering choices let a commodity microcontroller reliably deliver the raw physical-layer evidence (*stable CFO features*) required by AirCatch in realistic and dense BLE environments, while keeping the overall system deployable as a low-cost anti-tracking device.

[1] https://www.seeedstudio.com/Seeed-Studio-XIAO-MG24-p-6247.html Price: ≈$7
[2] https://www.alibaba.com/product-detail/ESP32-S3-Development-Board-Super-Mini_1601633352134.html Price: ≈$3

| Scenario | Dur. | #Packets | #MACs |
|---|---|---|---|
| Home→Work | 113.0 min | 18,216 | 6,251 |
| Work→Home | 104.8 min | 15,513 | 5,841 |
| Car commute | 63.3 min | 12,315 | 4,021 |
| Airport (no adv.) | 175.5 min | 8,905 | 2,467 |
| **Total** | **456.6 min** | **54,949** | **18,580** |

**Table 2: Dataset summary. The first three scenarios include an adversary, while the airport scenario is adversary-absent and used to stress-test false positives in dense public settings.**

*Power consumption.* We measured *BlePhasyr*'s energy use with a USB-C power meter over 5-minute intervals under three configurations. Full active operation (RF capture plus ESP32-S3 processing/forwarding) consumed 24 mWh (288 mW), the ESP32-S3 alone 14 mWh (168 mW), and continuous RF capture alone 4 mWh (48 mW), indicating that processing and transfer dominate the power budget. At full load, *BlePhasyr* requires 0.288 Wh/h, or 2.30 Wh over 8 hours of daily monitoring, equivalent to roughly 15% of a typical 4000 mAh smartphone battery. This is not a measurement of an integrated implementation; establishing the true energy cost, and reducing it where necessary, would require an optimized design integrated into the handset, which is left to future work.

*Deployment path and integration requirements.* The current *BlePhasyr* prototype is a wired USB-C peripheral, immediately useful as a research prototype and personal-safety accessory without modifying tracker ecosystems or smartphone operating systems. A near-term deployment path would integrate the MG24 and the associated transport logic onto a smaller board packaged as a USB-C dongle, a battery-pack accessory with anti-stalking capability, or a standalone receiver that communicates with the phone. Native smartphone integration would remove the external peripheral but requires chipset/OS support to expose raw IQ or equivalent fine-grained frequency features available through commodity application-level APIs.

## 7 Evaluation

This section presents a comprehensive evaluation of `AirCatch` across three dimensions: (i) *fingerprint validity*, including stability and discriminative power of the proposed CFO features; (ii) *baseline tracking detection* under naive adversarial behavior; and (iii) *robustness against evasive adversaries* that jointly manipulate transmission rate and identifier rotation. We further analyze the structure induced in the CFO feature space and evaluate the operational behavior of the resulting clustering and persistence-based detection pipeline.

### 7.1 Captured Datasets and Environments

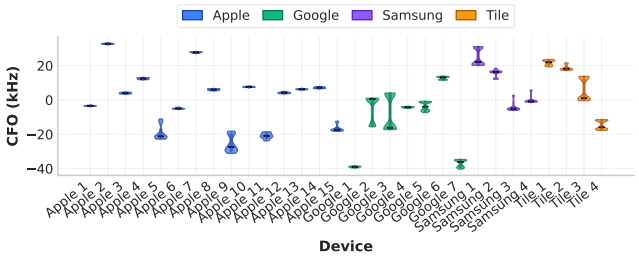
*Hardware.* Unless otherwise stated, we emulate adversarial trackers using custom ESP32 devices running modified OpenHaystack firmware. As detailed in Section 3.1, our evaluation includes fully functional ESP32-based trackers with firmware-controlled identifier rotation, as well as trace-based Apple adversaries (which used nRF hardware) and Google replay adversaries derived from captures of commercial tags. We selected ESP32 devices for functional attacker emulation because open-source tracker implementations were readily available and could be adapted to support aggressive identifier rotation with only minimal modifications.

We evaluate two receiver front-ends: (i) a *USRP B210* for high-fidelity IQ capture and offline CFO extraction, and (ii) *BlePhasyr*, which directly extracts embedded CFO features on-device. We report results separately for each receiver pipeline to quantify robustness across heterogeneous sensing stacks.

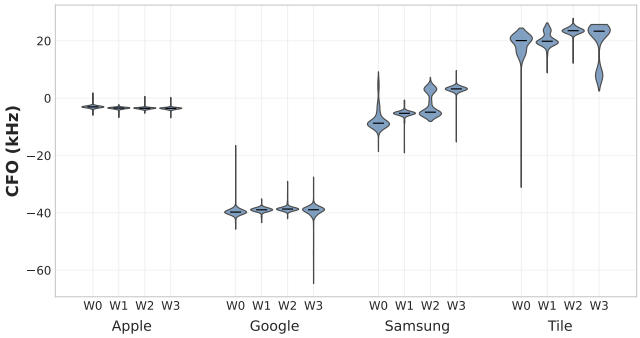
*Ethics and privacy.* Our data collection is designed to minimize privacy risk and to avoid collecting personally identifiable information (PII). We capture only over-the-air BLE advertising traffic (public beacons) and do not collect user content, account identifiers, location traces, or any auxiliary metadata beyond what is required to compute our RF features. We anonymize identifiers at ingestion: MAC addresses are *HMACed* and truncated using a fresh, per-scenario secret key, which is discarded after processing, preventing linkage across scenarios or datasets. For Samsung tags, we similarly HMAC-and-truncate the extracted PRIVID solely to obtain per-device ground truth during evaluation; we never retain raw PRIVIDs. Although packet bytes are parsed to locate fields needed for CFO extraction, we do not store raw payloads, and we do not retain any raw IQ samples: both payload and IQ are discarded immediately after feature computation, and only CFO-derived features are retained for analysis. We label as "adversary" only the devices we deploy and control for experiments; all other observed devices are treated as incidental bystanders/background. We conduct captures in public or controlled environments under our supervision and do not attempt to target specific individuals or to associate observed traffic with persons; any bystander traffic is processed only in anonymized, aggregate form. All released artifacts exclude raw identifiers, payloads, and IQ.

*Datasets.* We evaluate `AirCatch` on four real-world mobility traces captured using our low-cost *micro-SDR* (*BlePhasyr*) on BLE advertising channel 37. Each trace contains multiple mobility contexts, including walking, public transit, vehicular travel, and indoor stationary periods. We use the context boundaries recorded in the input traces to support phase-wise analysis. The first three traces correspond to *adversary-present* conditions in which our ESP32-based advanced tracker remains co-located with and follows the victim, whereas the airport trace is collected *without* an adversary and serves as a stress test for false positives in a dense public environment, particularly around baggage reclaim areas. Our experiments span laboratory, indoor, and outdoor environments with temperatures ranging from approximately $-10^{\circ}$C to $25^{\circ}$C. In total, our mobility evaluation spans 456.6 minutes, comprising 54,949 packets and 18,580 observed BLE advertiser addresses. Table 2 summarizes the datasets. Throughout the evaluation, we consider four representative commercial tracking ecosystems: Apple AirTag, Google Find My tags, Tile, and Samsung SmartTag.

*Adversarial scenario generation.* We evaluate an advanced adversary who explicitly attempts to evade detection by tuning its effective advertising period $T_{\text{tx}}$. In our threat model, the adversary *rotates its identifier at every transmission*; hence the identifier rotation period is *coupled* to the transmission period, i.e., $T_{\text{rot}} = T_{\text{tx}}$. This attacker is realistic and requires only firmware control over advertising rate and address randomization. We generate adversarial

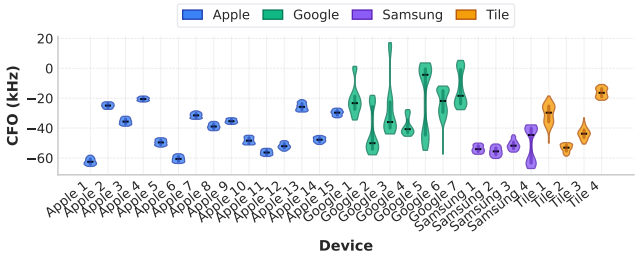



(a) USRP B210 (SDR): per-device CFO distributions for commodity tags. Each device exhibits a tight, device-specific CFO offset with clear inter-device separation.

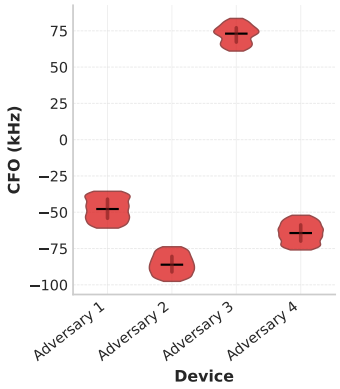



(b) USRP B210 (SDR): CFO over time (four 15-minute windows) for a representative tag of all categories. Consistency across windows indicates limited temporal drift during short periods.

**Figure 2: Stability of CFO-based fingerprints under SDR capture.**




(a) *BlePhasyr*: per-device CFO distributions for commodity tags. CFO offset widths and separations closely match the SDR in Figure 2a.




(b) *BlePhasyr*: CFO distributions for four ESP32-based advanced adversary trackers. Devices exhibit distinct CFO medians, supporting stable accumulation under per-transmission pseudonym rotation.

**Figure 3: *BlePhasyr* validation: CFO stability/separation on commodity tags, and separability of advanced adversary devices.**

datasets using our scenario generator by sweeping

$$T_{\text{tx}} \in \{2, 10, 15, 30, 60\}\ \text{s}, \qquad \text{with } T_{\text{rot}} = T_{\text{tx}}.$$

In the baseline traces of Table 2, both the adversary and benign tags transmit at a nominal advertising interval of 2s. For each $T_{\text{tx}}$ setting, we synthesize an evading adversary by replaying the selected tag's packets after (i) downsampling to the target transmission period and (ii) rotating its pseudonym on every emission. Varying $T_{\text{tx}}$ thus spans adversaries from highly active to stealthy.

*Stealth–utility trade-off.* This coupling induces a fundamental trade-off for the adversary. At *high transmission frequency* (small $T_{\text{tx}}$), the attacker emits more packets, which increases detection risk by producing denser and more persistent CFO-consistent evidence. However, frequent transmissions also improve tracking utility: more advertisements yield more opportunities for nearby third-party devices to observe the tracker and upload location reports, increasing the temporal granularity and reliability of victim localization. Conversely, in *stealth mode* (large $T_{\text{tx}}$), the attacker reduces its RF footprint and weakens the evidence available to the detector, but necessarily sacrifices tracking performance by lowering the rate of location updates.

We implemented an adversarial AirTag using a modified version of OpenHaystack [15]. The ESP32-based tracker successfully integrated into both Apple and Google tracking ecosystems, enabling location retrieval through the corresponding crowdsourced networks. The device transmits a single packet before rotating its MAC address every two seconds, thereby emulating an aggressive identifier-rotating adversary. We subsequently queried Apple's Find My network for corresponding reports and observed successful detections for most ephemeral identifiers. It demonstrates that even a single transmitted packet per MAC address can remain sufficient for practical tracking.

## 7.2 Effectiveness of Chosen Fingerprints

We first validate that the proposed CFO-derived fingerprints satisfy two fundamental requirements for robust tracking detection: *(i) temporal stability* for a given physical transmitter across time and mobility contexts, and *(ii) discriminative power* across distinct devices,

such that packets emitted under rapidly changing pseudonyms still accumulate into a compact and device-consistent CFO region. We evaluate these properties on both (i) a high-fidelity USRP B210 SDR pipeline and (ii) our low-cost *BlePhasyr* platform, which extracts the same packet-level and transition-conditioned CFO features directly on-device.

### 7.2.1 *Stability across time and sensing stacks.*

*Protocol.* We collect approximately one-hour traces containing multiple commodity tags from each ecosystem operating under nominal behavior. For each packet, we extract the holistic CFO fingerprint and aggregate per-device feature distributions. To evaluate temporal stability, we partition traces into consecutive 15-minute windows, matching the temporal granularity later used by the detection pipeline, and compare the resulting per-window CFO distributions.

*CFO stability under high-fidelity SDR capture.* Figure 2a shows per-device CFO distributions extracted using the USRP B210. Each device occupies a compact and highly device-specific CFO region, while different devices remain well separated in feature space. Across all devices, CFO values remain confined to a narrow tens-of-kHz range (approximately $[-40, +40]$ kHz in this experiment), and the within-device dispersion is substantially smaller than the inter-device separation. This separation regime is precisely what enables stable clustering and persistence analysis under identifier rotation.

*Limited temporal drift.* Figure 2b reports the same devices partitioned into four consecutive 15-minute windows. The resulting CFO distributions remain highly stable across time: medians exhibit minimal displacement, and intra-window spreads overlap substantially. This bounded temporal drift justifies both the feature space used by `AirCatch` and the long-horizon persistence analysis employed for tracker detection.

*Agreement between BlePhasyr and SDR measurements.* Figure 3a demonstrates that *BlePhasyr* preserves the same qualitative CFO geometry observed with the SDR pipeline. Device-specific CFO offsets remain concentrated within a narrow frequency band while maintaining clear inter-device separation, validating that low-cost embedded extraction preserves the structure required for tracking detection. Figure 3 (right) further reports the CFO distributions of our ESP32-based advanced adversarial trackers as observed by *BlePhasyr*. The adversarial devices occupy distinct and compact CFO regions, with medians separated by multiple kHz to tens of kHz, allowing packets emitted under rapidly rotating identifiers to accumulate into persistent high-density clusters rather than dispersing into environmental background traffic.

### 7.2.2 *Discriminative Nature of CFO Features.*

*Transition-conditioned CFO features provide additional geometric structure.* While a single packet-level CFO offset already provides some device separation, crowded environments and partially overlapping CFO medians motivate richer physical-layer representations such as transition-conditioned CFO features. Figure 5 shows that these transition-specific distributions remain highly consistent for a given device while introducing complementary separation axes across devices. As a result, packets emitted under rapidly rotating identifiers remain concentrated within the same joint CFO

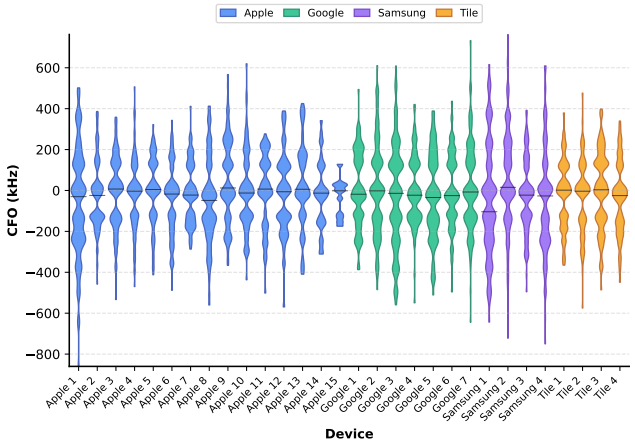



**Figure 4: CFO estimates from prior CFO-based fingerprinting [12]. Outputs show high variance, contrasting with the stable tens-of-kHz regime of our CFO/transition-CFO pipeline.**

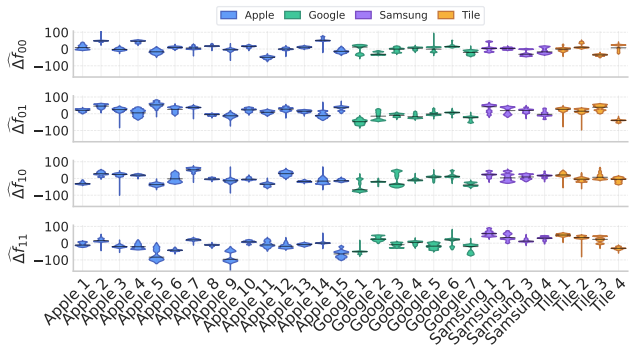



**Figure 5: Transition-CFO fingerprints ($\widehat{\Delta f}_{00}$, $\widehat{\Delta f}_{01}$, $\widehat{\Delta f}_{10}$, $\widehat{\Delta f}_{11}$) in kHz for commodity tags. Transition-conditioned CFO features remain stable per device and provide additional separation beyond a single CFO offset.**

embedding, whereas unrelated background devices remain more dispersed.

*Supervised fingerprinting validates separability across sensing stacks.* To quantify discriminative power, we perform within - ecosystem device identification using only CFO-derived fingerprints. Specifically, we train a Random Forest classifier to predict device identity from packet-level and transition-conditioned CFO features using an 80:20 temporal split over a one-hour capture, ensuring that training and testing operate on disjoint time intervals and preventing temporal leakage. Both the SDR pipeline and *BlePhasyr*achieve an accuracy of 85% and an F1-score of 83%, demonstrating that the low-cost embedded sensing stack preserves the discriminative power of the proposed CFO-derived fingerprints.

*Feature ablation.* Using the same classifier and temporal split, we compare the complete five-dimensional fingerprint against packet-level CFO alone. On the SDR pipeline, CFO alone achieves 78.70%

accuracy and a 75.99% F1-score, representing reductions of 6.30 and 7.01 percentage points, respectively, relative to the complete fingerprint. On the *BlePhasyr* pipeline, CFO-only performance drops to 26.22% accuracy and a 23.76% F1-score. Adding the transition-conditioned components raises *BlePhasyr*'s performance to 85% accuracy and an 83% F1-score, matching the discriminative performance of the high-fidelity SDR pipeline. This shows that the additional features compensate for the limitations of the low-cost receiver and are essential for reliable device separation on *BlePhasyr*.

*Comparison with prior CFO-based fingerprinting.* We additionally compare against a prior CFO-based fingerprinting pipeline [12]. Due to runtime constraints, the comparison was performed on a subset of 5,000 packets, for which the baseline produced valid CFO estimates for approximately 3,400 packets. Figure 4 shows that the resulting CFO estimates exhibit substantially higher variance, spanning several hundreds of kHz (approximately $[-800, +600]$ kHz in our experiments), and significantly lower temporal stability than the proposed CFO and transition-CFO pipeline (Figures 2, 3, and 5). These results reinforce a key design principle of `AirCatch`: prioritizing lightweight and operationally stable physical-layer fingerprints that can be continuously extracted on resource-constrained hardware such as *BlePhasyr*, rather than high-variance estimators that are difficult to deploy reliably in real-world tracking detection scenarios.

## 7.3 Adversary Detection by State-of-the-Art

*Detection infrastructure.* We evaluate tracker detectability using both platform-native protections and third-party tooling. Concretely, we carry an Apple iPad Pro (M1) running iPadOS 16.2 (iOS unwanted-tracker alerts) and a Google Pixel 7 Pro running Android 16 (Android unknown-tracker alerts), and we additionally run *AirGuard* on the both of these devices to provide cross-ecosystem coverage. This setup lets us compare (i) Apple's and Google's built-in mechanisms against (ii) a widely used third-party detector. We note that iOS restricts third-party access to BLE MAC addresses, limiting observability for non-system applications and thereby constraining third-party detectors on iOS.

*Threat model and expectation.* We consider an adversary that deploys a *single physical tracker* while frequently rotating protocol-visible identifiers (e.g., MAC address, public key material, or both depending on ecosystem constraints). Existing deployed defenses largely depend on correlating identifiers and advertisement metadata over time; thus, sufficiently fast rotation and/or sparse transmissions should reduce linkability and delay (or prevent) alerts. Our goal here is not to exhaustively benchmark OS implementations, but to characterize whether current detectors remain effective under realistic evasion patterns used by advanced trackers.

*Naive trackers already incur non-trivial alert latencies.* In repeated trials where a detector supports the tracker ecosystem, we observe alert latencies of at least 30 minutes and sometimes exceeding one hour. Detection generally occurs faster during nighttime conditions. Among the evaluated systems, AirGuard consistently generated alerts sooner than both Apple and Google while maintaining broader ecosystem coverage. Apple devices detected AirTag variants and also detected Google-based trackers. In contrast, Google devices detected original AirTags and Google tags but did not detect other Find My–compatible trackers from third-party manufacturers. Trackers from Samsung and Tile were not detected by either Apple or Google in our tests. AirGuard, however, detected Apple, Google, Samsung, and Tile trackers, typically within a similar 30-minute window.

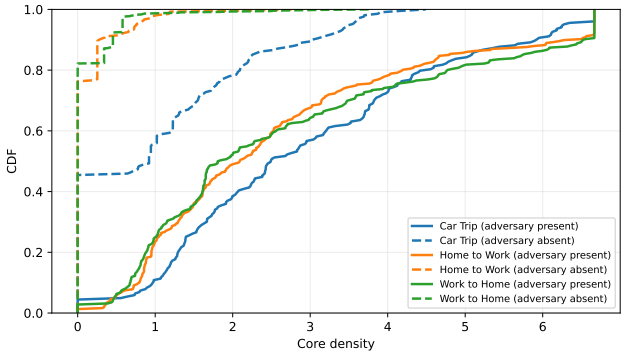



**Figure 6: Core-density distribution shift under tracking. Empirical CDF of cluster core densities aggregated per route, overlaying adversary-present vs adversary-absent traces.**

*Advanced identifier rotation and stealth evade existing defenses.* Our experiments show that existing anti-tracking mechanisms fundamentally struggle against aggressive identifier rotation and stealthy transmission behavior. In particular, adversaries employing fast identifier rotation across all evaluated schemes (from 2 s to 60 s rotation intervals) are consistently missed by current defenses. Further, when the adversary operates in a stealthy low-duty-cycle mode by transmitting only once per minute while keeping the identifier constant, notifications are either missed entirely or generated only after long (multi-hour) operational delays.

*Protocol constraints do not eliminate the threat.* Apple's Find My tightly couples public key material to the BLE address in advertisement processing, which restricts certain *MAC-only* or *key-only* rotation modes. However, this coupling does not guarantee timely detection in practice: under combined identifier changes permitted by the ecosystem, built-in mechanisms still fail to flag a co-moving tracker. Conversely, ecosystems where MAC and identifier rotation are more decoupled inherently expose a broader evasion surface, enabling even advanced rotation strategies that directly destroy identifier-based correlation.

## 7.4 Adversary detection by `AirCatch`

We set the persistence requirement to $T_{\min} = 40$ minutes as a practical trade-off between early detection and false-positive robustness. This value is chosen to provide timely alerts to users while ensuring that benign co-location events do not accumulate sufficient evidence to trigger detection. Lower thresholds enable faster responses but increase the risk of transient dense clusters producing false positives, whereas higher thresholds further suppress false alarms at the cost of delayed adversary detection. We find 40 minutes to strike a favorable balance between responsiveness and reliability.

Across multiple scenarios (three real mobility commutes with an adversary present, plus one adversary-absent airport trace),

| Scenario | Benign trace (adv-absent) | ESP32 adv. (1–4) | Apple replay adv. (1) | Google replay adv. (1) | $T_{tx}$=$T_{rot}$ settings |
|---|---|---|---|---|---|
| Airport (stress test; no adv.) | ✓ | – | – | – | – |
| Home→Work (adv present) | ✓ | ✓ | ✓ | ✓ | {2, 10, 15, 30, 60} s |
| Work→Home (adv present) | ✓ | ✓ | ✓ | ✓ | {2, 10, 15, 30, 60} s |
| Car commute (adv present) | ✓ | ✓ | ✓ | ✓ | {2, 10, 15, 30, 60} s |

**Table 3: `AirCatch`'s tracker detection. Each row is a scenario; the first column reports the *captured benign scenario* (i.e., no adversary present). The remaining columns report whether `AirCatch` correctly flags adversary-present blocks when we inject (i) 1–4 distinct *ESP32* trackers (each rotating its pseudonym *at every transmission*), and (ii) "replay" adversaries created from *commodity Apple/Google tags* by downsampling to a target transmission period and enforcing per-transmission pseudonym rotation. ✓ denotes correct behavior (no false alarms on benign traces; correct flag when an adversary is present).**

`AirCatch` cleanly separates background lost-state tag traffic from advanced tracking behavior.

Concretely, for *all* benign captures (Home→Work, Work→Home, Car commute, and Airport stress-test), the detector remains negative throughout, despite high tag-traffic and rapid context transitions (walking, transit, indoor stops).

We deploy an advanced attacker that *rotates its pseudonym at every transmission* and varies its transmission period $T_{tx}$, spanning from highly active ($T_{tx}$=2s) to stealthy ($T_{tx}$=60s) operation. We evaluate (i) 1–4 concurrent ESP32-based advanced trackers (distinct physical devices), and (ii) replayed Apple and Google tags made to emulate the same *per-transmission* rotation and downsampled transmission schedule using our scenario generator.

Table 3 summarizes `AirCatch`'s detection outcomes. Whenever an adversary is present, `AirCatch` identifies a *persistently compact* CFO core whose density exceeds the operational threshold $\delta$ and whose persistence exceeds $T_{min}$, while background clusters remain short-lived and are typically below $\delta$. At the selected $T_{min}$=40 min operating point, `AirCatch` detects all adversary-present configurations and remains negative on all four benign scenarios, yielding a 100% true-positive rate, 0% false-positive rate, 0% false-negative rate, and 100% true-negative rate. Since an alert is emitted only after the persistence criterion is satisfied, the alert latency is 40 min.

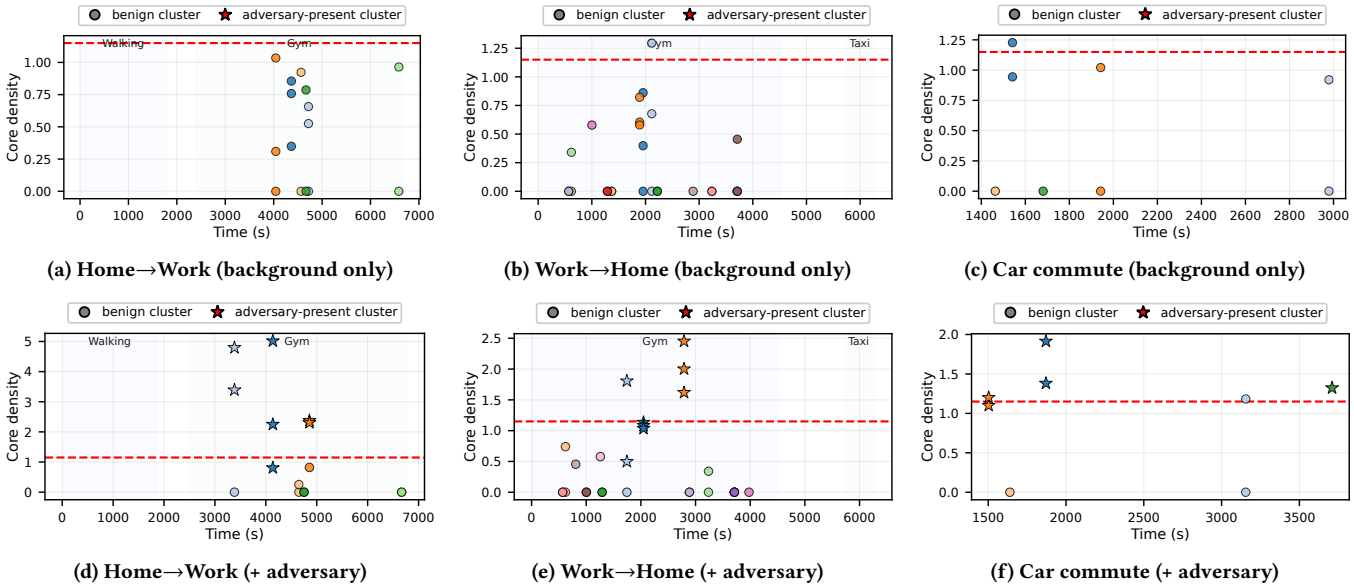
*Sensitivity to the persistence threshold.* We sweep $T_{min}$ from 10 to 60 min across all scenario/configuration combinations in Table 3 (more than 100 total cases). Across this entire range, both the false-positive and missed-detection rates individually remain below 4%, demonstrating the robustness of the `AirCatch` detection pipeline to the choice of persistence threshold. Shorter thresholds reduce the evidence required for an alert but increase sensitivity to transient benign activity: at 10 min, three false alerts occur (2.52% of all cases). This decreases to 1.68% at 20 min, with no missed adversarial detections. At $T_{min}$ = 40 min, `AirCatch` correctly classifies all evaluated cases, yielding 0% false alerts and 0% missed detections. Increasing the threshold does not consistently improve performance: 50 min yields 1.68% false alerts and 1.68% missed detections; and at 60 min the false-alert rate rises to 2.52%. Thus, 40,min is the best observed operating point balancing delay, false alerts, and missed detections.

## 7.5 Behavior of Core Densities

*Sensitivity to the core-density threshold $\delta$.* We stress-test the decision threshold by sweeping $\delta$ around our chosen operating point ($\delta$=1.15) and observing the transition from *over-sensitive* (false positives) to *over-strict* (false negatives) regimes. When $\delta$ is reduced below the selected value, the detector begins to admit benign compact clusters: at $\delta$=0.9 we observe positives in benign settings (e.g., Airport and Car commute), while true adversaries remain detectable. At $\delta$=1.0, the Airport false positive persists, confirming that too-low thresholds primarily hurt precision without improving recall. In contrast, increasing $\delta$ makes the rule conservative and starts to miss adversary blocks: at $\delta$=1.3 we already incur false negatives in Car commute, and at $\delta$=1.45 the degradation is stronger (e.g., in Work to home and in Car commute). These results validate $\delta$=1.15 as an *optimal knee point*: it is the smallest value that suppresses the observed false positives (restoring precision) while still retaining full recall (no false negatives) across our evaluated adversary settings. Practically, $\delta$ provides a user-tunable robustness knob: selecting $\delta$>1.15 can further harden against rare false positives, but necessarily increases the risk of missed detections for sparse trackers.

*Core-density distribution.* Figure 6 shows a pronounced distributional separation between background-only and adversary-present traces. At our operational threshold $\delta$ = 1.15, the *background-only* CDF already saturates for two routes: 98.43% (Work→Home) and 98.78% (Car) of cluster cores fall at or below $\delta$, i.e., only 1.57% and 1.22% exceed the threshold, respectively. Home→Work is the most challenging background route (denser ambient RF activity), yet still 59.18% of cores remain $\leq \delta$ and its high-density tail is handled by the persistence constraint (near-threshold clusters are short-lived and fail $T_{min}$). In sharp contrast, in *adversary-present* traces only 14.15% (Home→Work), 27.18% (Work→Home), and 29.89% (Car) of cores lie below $\delta$, meaning that 70–86% of cores exceed the density threshold when tracking is present. This shift is not a small "margin" effect: the median core density moves from $\leq 0.92$ in background-only traces to $\approx$ 1.81–2.49 under tracking, and the upper tail expands dramatically (e.g., 95th-percentile core density reaches 6.27–6.67 with an adversary, versus 0.58–0.76 in two of the three background routes). Overall, the CDF confirms that core density is a robust route-agnostic discriminator: benign co-location concentrates at low densities, whereas tracking reliably induces a heavy high-density tail, enabling a low-FP operating point when combined with persistence ($T_{min}$).

*Why core density separates benign co-location from tracking.* Figure 7 illustrates the temporal evolution of core density for clusters produced by our pipeline across three real mobility traces (Home→Work, Work→Home, and car commute), each segmented into contextual phases (e.g., walking, gym, taxi). In the *background-only* traces (top row), cluster densities remain consistently low and short-lived: Home→Work never exceeds a density of 1.0, while the




(a) Home→Work (background only) (b) Work→Home (background only) (c) Car commute (background only)

(d) Home→Work (+ adversary) (e) Work→Home (+ adversary) (f) Car commute (+ adversary)

**Figure 7: Core density over time for three mobility scenarios, shown for (top) background-only traces and (bottom) traces with an ESP32-based advanced adversary (stealth, transmitting per minute with new identifier) following the user. Adversary is flagged only if clusters' density exceeds $\delta = 1.15$ (shown by the dashed red line) and they persist for at least $T_{\min} = 40$ minutes.**

remaining commutes fluctuate within a narrow band below 1.25, with the vast majority of clusters falling beneath our operational threshold $\delta = 1.15$. This behavior reflects benign RF environments in which clusters correspond to distinct devices with heterogeneous CFOs and limited temporal overlap, yielding sparse and transient structure rather than a repeatedly accumulating CFO core. Potential inflation from continuously transmitting stationary devices is mitigated by our windowed aggregation strategy (Algorithm 1), which prevents packet-level repetition from artificially increasing cluster density.

We further quantify this effect by examining the persistence of dense CFO clusters (core density $> \delta$) under benign and adversarial conditions. In benign scenarios, only a small number of clusters ever exceed the density threshold, with $n$=10 in the airport stress trace, $n$=86 during the car commute, $n$=5 for Home→Work, and $n$=9 for Work→Home. In contrast, adversarial scenarios yield substantially more dense clusters, with $n$=175, $n$=224, and $n$=201 clusters in the car commute, Home→Work, and Work→Home traces, respectively. Moreover, adversarial clusters persist for significantly longer durations, exhibiting median persistence between 1570 s and 2120 s, often spanning most or all of a detection block, whereas benign clusters remain short-lived, with medians well below this range (e.g., 760 s and 1180 s). Consequently, benign activity rarely accumulates sufficient sustained evidence across blocks to trigger detection, while adversarial trackers consistently form long-lived dense cores that surpass the persistence threshold.

Consistent with this quantitative behavior, when an ESP32-based evasive adversary is present (bottom row), transmitting intermittently and rotating ADV addresses, the adversary-associated cluster rapidly forms an abnormally compact CFO core whose density rises well above $\delta$ and remains elevated over time. In Home→Work, density reaches values up to 5, while Work→Home and the car commute reach approximately 2.5 and 2.0, respectively, reflecting repeated accumulation of short-lived identifiers within the same CFO-consistent region. While a small number of benign clusters may briefly approach $\delta$ during high-density mobility phases, these fail to satisfy the persistence constraint and decay before $T_{\min} = 40$ minutes. Together, the conjunction of density and persistence enables no observed false alerts while preserving sensitivity to trackers that actively evade linkability by rotating and intermittently transmitting.

## 8 Conclusion

With the proliferation of tag-based ecosystems, advanced trackers increasingly get the opportunity of evading current OS-level protections by rotating identifiers and duty-cycling transmissions. We introduce AirCatch, which shows that analog-layer invariants can restore reliable tracking detection even when the digital surface is adversarially unstable. We present the first end-to-end defense that extracts stable CFO fingerprints from commodity BLE advertisements, maps them into robust embeddings, and detects active evasion by flagging persistent, abnormally compact CFO cores that repeatedly accumulate short-lived identifiers across time, shifting detection from brittle identifier logic to a physically grounded signature that is difficult to erase without sacrificing radio functionality. Beyond a single system, AirCatch contributes a practical, reproducible foundation for the community: a low-cost BLE SDR named *BlePhasyr*, a principled tracker algorithm that separates benign tags from advanced trackers, and an easy-to-use anti-tracking application. As tag-based stalking scales, AirCatch advances the state of the art by closing the gap between current assumptions and real adversarial behavior.

## Ethical Considerations

Our study examines the detectability of commodity BLE trackers and introduces `AirCatch`, a receiver-side detector operating solely on broadcast BLE advertisements observed in the environment. We collected traffic in four real-world mobility traces and in controlled experiments using our own ESP32-based trackers, strictly observing bystander devices without interaction, connection, targeting, or attempts to identify individuals, and limiting measurements to public or semi-public spaces reflecting ordinary mobility conditions. To minimize privacy risk, we do not retain raw identifiers in any released artifacts: all MAC addresses and ecosystem-specific identifiers (e.g., Samsung PRIVIDs) are keyed-hashed and truncated using per-capture keys that are destroyed after processing, preventing cross-dataset linkage or re-identification. Advertisement payloads and raw IQ samples are used only transiently to compute physical-layer features (e.g., CFO-derived statistics) and are discarded thereafter; downstream analysis retains only derived, non-content features and aggregate statistics, without storing user identities or application-layer data. Moreover, CFO is generally insufficient as a robust long-term fingerprint and has primarily been used in prior work for short-term tracking or device identification. In our setting, CFO estimates are approximate and intended only for short-term clustering and persistence analysis under identifier rotation, reducing the feasibility of long-term device linking. We therefore do not believe this work materially introduces new long-term dual-use risks. Instead, our low-cost platform enables practical and user-accessible defensive research on unwanted BLE tracking.

## Open Science

We make all artifacts required to evaluate and reproduce the contributions of this paper available through an anonymous, public repository:

`https://github.com/miishra/AirCatch/tree/main`

The repository is fully self-contained and includes the complete implementation of our end-to-end pipeline, scripts to reproduce the main experiments, and comprehensive documentation describing how to execute each stage. In accordance with the Open Science Policy, the artifact provides a detailed README along with all code necessary for data capture across supported platforms, the detection algorithm, and the accompanying mobile application, enabling independent validation and reuse by the community.

## AI Use

The authors used generative AI-based tools to assist with language refinement, including improving text flow and correcting typographical, grammatical, and stylistic issues. All technical content, experiments, analyses, and conclusions were developed and verified by the authors.

## Acknowledgments

This material is based upon work partially supported by the US National Science Foundation under Grant No. 2434016.

## References

[1] Abdulsahib Albehadili and Ahmad Y Javaid. 2024. Performance Evaluation of Carrier-Frequency Offset as a Radiometric Fingerprint in Time-Varying Channels. *Sensors* 24, 17 (2024), 5670.

[2] Johannes K Becker, David Li, and David Starobinski. 2019. Tracking anonymized bluetooth devices. *Proceedings on Privacy Enhancing Technologies* (2019).

[3] Leon Böttger, Alexander Matern, Dennis Arndt, and Matthias Hollick. 2025. Okay Google, Where's My Tracker? Security, Privacy, and Performance Evaluation of Google's Find My Device Network. *Proceedings on Privacy Enhancing Technologies* (2025).

[4] Jimmy Briggs and Christine Geeng. 2022. Ble-doubt: Smartphone-based detection of malicious bluetooth trackers. In *2022 IEEE Security and Privacy Workshops (SPW)*. IEEE, 208–214.

[5] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. 2008. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*. 116–127.

[6] Hanlin Cai, Yuchen Fang, Jiacheng Huang, Honglin Liao, Meng Yuan, and Zhezhuang Xu. 2024. Securing Billion Bluetooth Low Energy Devices Using Cyber-Physical Analysis and Deep Learning Techniques. (2024).

[7] Rose Ceccio, Sophie Stephenson, Varun Chadha, Danny Yuxing Huang, and Rahul Chatterjee. 2023. Sneaky spy devices and defective detectors: the ecosystem of intimate partner surveillance with covert devices. In *32nd USENIX Security Symposium (USENIX Security 23)*. 123–140.

[8] Junming Chen, Xiaoyue Ma, Lannan Luo, and Qiang Zeng. 2025. Tracking you from a thousand miles away! turning a bluetooth device into an apple {AirTag} without root privileges. In *34th USENIX Security Symposium (USENIX Security 25)*. 4345–4362.

[9] Tess Despres, Noelle Davis, Prabal Dutta, and David Wagner. 2023. DeTagTive: Linking MACs to protect against malicious BLE trackers. In *Proceedings of the Second Workshop on Situating Network Infrastructure with People, Practices, and Beyond*. 1–7.

[10] Harry Eldridge, Gabrielle Beck, Matthew Green, Nadia Heninger, and Abhishek Jain. 2024. {Abuse-Resistant} Location Tracking: Balancing Privacy and Safety in the Offline Finding Ecosystem. In *33rd USENIX Security Symposium (USENIX Security 24)*. 5431–5448.

[11] Dañiel Gerhardt, Matthias Fassl, Carolyn Guthoff, Adrian Dabrowski, and Katharina Krombholz. 2025. AirTag-Facilitated Stalking Protection: Evaluating Unwanted Tracking Notifications and Tracker Locating Features. (2025).

[12] Hadi Givehchian, Nishant Bhaskar, Eliana Rodriguez Herrera, Héctor Rodrigo López Soto, Christian Dameff, Dinesh Bharadia, and Aaron Schulman. 2022. Evaluating physical-layer ble location tracking attacks on mobile devices. In *2022 IEEE symposium on security and privacy (SP)*. IEEE, 1690–1704.

[13] Mohamed I Hany, Hamada Rizk, and Moustafa Youssef. 2024. Airtags for human localization, not just objects. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Geo-Privacy and Data Utility for Smart Societies*. 13–18.

[14] 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*. 26–38.

[15] 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*. 374–376.

[16] Alexander Heinrich, Leon Würsching, and Matthias Hollick. 2024. Please Unstalk Me: Understanding Stalking with Bluetooth Trackers and Democratizing Anti-Stalking Protection. *Proceedings on Privacy Enhancing Technologies* (2024).

[17] Hazem Ibrahim, Rohail Asim, Matteo Varvello, and Yasir Zaki. 2023. I tag, you tag, everybody tags!. In *Proceedings of the 2023 ACM on Internet Measurement Conference*. 561–568.

[18] HyunSeok Daniel Jang, Hazem Ibrahim, Rohail Asim, Matteo Varvello, and Yasir Zaki. 2025. A Tale of Three Location Trackers: AirTag, SmartTag, and Tile. *arXiv preprint arXiv:2501.17452* (2025).

[19] Akshaya Kumar, Anna Raymaker, and Michael Specter. 2025. Security and Privacy Analysis of Tile's Location Tracking Protocol. arXiv:2510.00350 [cs.CR] https://arxiv.org/abs/2510.00350

[20] Xiaofeng Liu, Chaoshun Zuo, Qinsheng Hou, Pengcheng Ren, Jianliang Wu, Qingchuan Zhao, and Shanqing Guo. 2025. A Thorough Security Analysis of {BLE} Proximity Tracking Protocols. In *34th USENIX Security Symposium (USENIX Security 25)*. 5347–5364.

[21] 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. *arXiv preprint arXiv:1904.10600* (2019).

[22] 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* (2023).

[23] Katharina OE Müller, Louis Bienz, Bruno Rodrigues, Chao Feng, and Burkhard Stiller. 2023. Homescout: Anti-stalking mobile app for bluetooth low energy devices. In *2023 IEEE 48th Conference on Local Computer Networks (LCN)*. IEEE, 1–9.

[24] Katharina OE Müller, Samuel Frank, Dario Monopoli, Daria Schumm, Bruno Rodrigues, and Burkhard Stiller. 2025. Smart Shielding: Using ML and AirTag RSSI Data for Better Privacy. In *2025 IEEE 50th Conference on Local Computer Networks (LCN)*. IEEE, 1–7.

[25] Katharina OE Müller, Stefan Saxer, Bruno Rodrigues, et al. 2025. AirTagged: A Dataset and Processing Framework for Heterogeneous High-Density IoT Environments. (2025).

[26] Ali Nikoofard, Hadi Givehchian, Nishant Bhaskar, Aaron Schulman, Dinesh Bharadia, and Patrick P Mercier. 2022. Protecting bluetooth user privacy through obfuscation of carrier frequency offset. *IEEE Transactions on Circuits and Systems II: Express Briefs* 70, 2 (2022), 541–545.

[27] Lauren R Pace, LaSean A Salmon, Christopher J Bowen, Ibrahim Baggili, and Golden G Richard III. 2023. Every step you take, I'll be tracking you: Forensic analysis of the tile tracker application. *Forensic science international: digital investigation* 45 (2023), 301559.

[28] 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, 301–311.

[29] Narmeen Shafqat, Nicole Gerzon, Maggie Van_Nortwick, Victor Sun, Alan Mislove, and Aanjhan Ranganathan. 2023. Track you: A deep dive into safety alerts for apple airtags. *Proceedings on Privacy Enhancing Technologies* 2023, 4 (2023).

[30] George-Alexandru Stoian, Thiemo Voigt, and Christian Rohner. 2025. Augmenting BLE Fingerprinting Using Instantaneous Frequency. In *18th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 274–279.

[31] Weiping Sun, Jeongyeup Paek, and Sunghyun Choi. 2017. CV-track: Leveraging carrier frequency offset variation for BLE signal detection. In *Proceedings of the 4th ACM Workshop on Hot Topics in Wireless*. 1–5.

[32] Kieron Ivy Turk and Alice Hutchings. 2024. Stop Following Me! Evaluating the Malicious Uses of Personal Item Tracking Devices and Their Anti-Stalking Features. In *Proceedings of the 2024 European Symposium on Usable Security*. 277–289.

[33] Jianliang Wu, Yuhong Nan, Vireshwar Kumar, Mathias Payer, and Dongyan Xu. 2020. {BlueShield}: Detecting spoofing attacks in bluetooth low energy networks. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*. 397–411.

[34] Tingfeng Yu, James Henderson, Alwen Tiu, and Thomas Haines. 2024. Security and Privacy Analysis of Samsung's {Crowd-Sourced} Bluetooth Location Tracking System. In *33rd USENIX Security Symposium (USENIX Security 24)*. 5449–5466.

# A Behavior of Commodity Trackers

We analyze the protocol behavior and privacy mechanisms of widely deployed BLE-based tracking ecosystems: Apple, Samsung, Google, and Tile, focusing on how they randomize link-layer identifiers, rotate protocol-visible cryptographic material, and trigger protections against unwanted tracking. Our analysis is organized along four dimensions: (i) MAC address randomization, (ii) public key / identifier updates, and (iii) stalker detection mechanisms. The corresponding packet layouts that realize these design choices are illustrated in Figure 8

*System model and terminology.* Across ecosystems, we distinguish three roles: (i) the *owner device*, which is authenticated and associated with the tracker; (ii) *helper devices*, which periodically scan BLE advertisements and upload encrypted location reports; and (iii) the *tracked device* (the BLE tag). Helper devices do not learn the identity of the tag owner and act solely as relays to the backend infrastructure.

## A.1 Samsung Find Network

Samsung's Find Network [34] consists of an owner device (a Samsung smartphone), helper devices (nearby Samsung devices), and trackers. Communication between the owner and the tag relies on BLE advertisements and a proprietary GATT profile. SmartTag advertises using Samsung-specific service data formats.

**Operational states.** SmartTag operates in multiple privacy-relevant states: *connected*, *premature lost*, *lost*, and *overmature lost*. After a separation interval, the tag transitions into a "lost" reporting regime to enable relay by helper devices; after long separation, it enters an overmature regime where protocol activity is reduced for power efficiency and to support anti-tracking workflows.

**Identifier and MAC address rotation.** SmartTags periodically updates a privacy identifier (PRIVID) and auxiliary fields embedded in BLE service data. The PRIVID remains constant in the lost state, for 24 hours, but notably, SmartTags continue to randomize its BLE MAC address even in lost states, potentially reducing long-term linkability at the link layer if one is only using MAC address.

## A.2 Apple Find My Network

Apple's Find My network [14, 29] consists of Apple owner devices logged into iCloud, helper devices running iOS or macOS, and tracked devices such as AirTags. AirTags rely exclusively on BLE advertisements and do not establish persistent connections with helper devices.

**Protocol design.** AirTags periodically broadcast advertisement payloads containing rotating identifiers derived from ephemeral public keys. The corresponding private keys are stored only on the owner's device, ensuring helper devices and the backend cannot decrypt location information.

**State transitions and safety.** AirTags transition between near-owner and separated/lost modes based on proximity and time. Advertisement identifiers rotate every 15 minutes in the connected state but stay constant in the lost state; Apple complements cryptographic unlinkability with OS-level anti-stalking defenses (user notifications and audible alerts). Prior work shows that while these mechanisms provide strong unlinkability guarantees, residual tracking risks remain under certain conditions [14, 29]. On the other side, iPhones and Airpods that transmit packets always change their MAC address every 15 minutes. Airpods transmit packets in connected and lost state. Meanwhile, phone transmits when not connected to the network.

## A.3 Google Find My Device Network

Google's Find My Device network [3] provisions trackers using Fast Pair and uses public-key material to encrypt location reports relayed by helper devices. BLE advertisements use an ecosystem-specific service format.

**Unwanted Tracking (UT) mode.** Google introduces a server-controlled Unwanted Tracking (UT) mode that is expected to activate after prolonged separation from the owner and the completion of helper participation. In UT mode, Google trackers do not rotate their Bluetooth MAC address; instead, they retain a fixed address while rotating the cryptographic identifier in the advertising payload approximately every 15 minutes. Moreover, prior work [3] showed that UT activation depends on a server-side timer, which can be reset by submitting a forged "owner present" report, thereby preventing UT mode from triggering. In practice, this design exposes meaningful degrees of freedom for firmware-controlled adversaries, who can balance stealth (e.g., low duty cycle and delayed UT activation) against localization capability.

## A.4 Tile Network

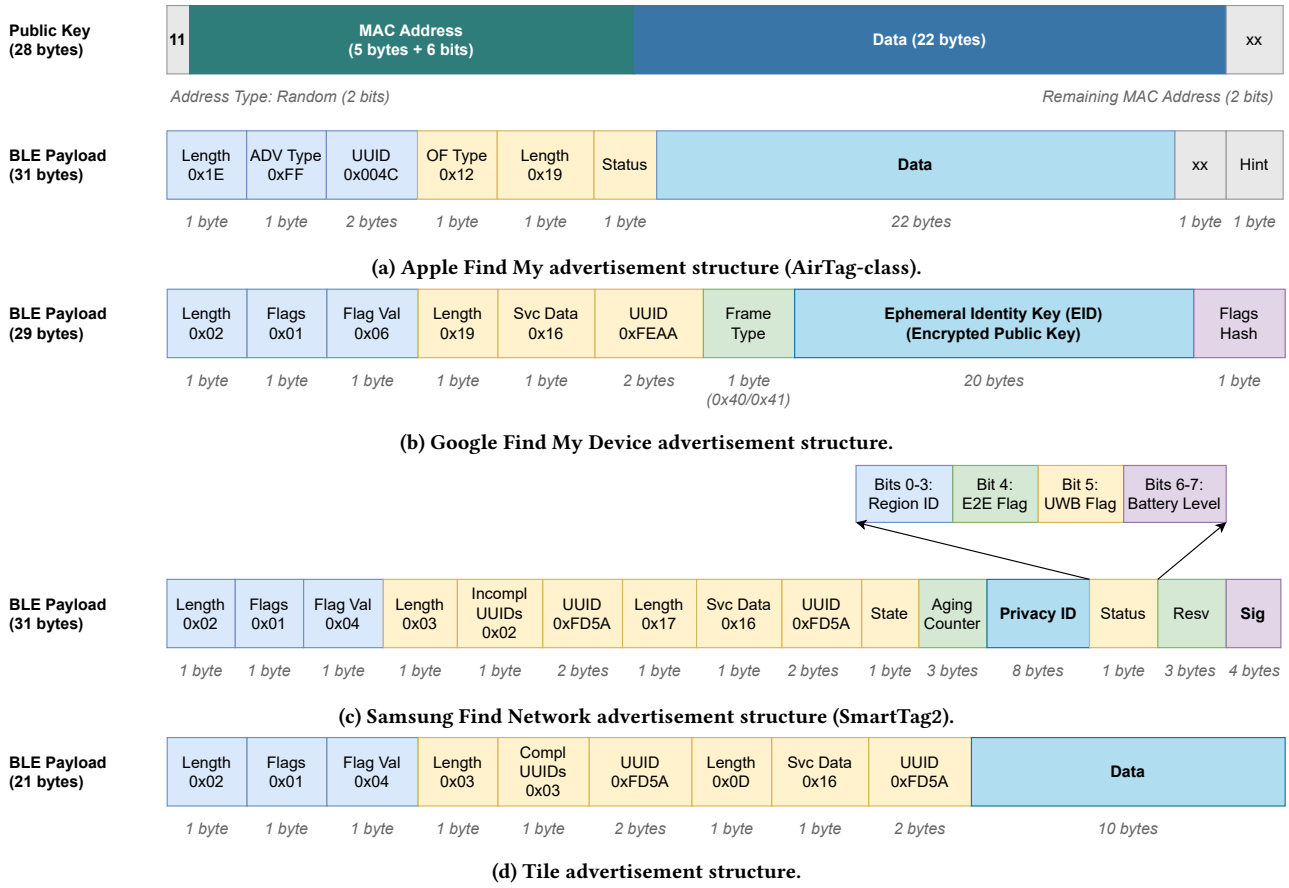
Tile's ecosystem [19] consists of Tile trackers, owner devices, and helper devices running the Tile application. Unlike Apple, Google, and Samsung, Tile relies on application-level participation rather than OS-level integration, resulting in more limited helper coverage.

**Identifier stability and anti-stalking.** Tile trackers broadcast BLE advertisements containing identifiers that remain stable for extended periods. It does not change its MAC address in connected or lost states, enabling longer-lived linkability. Anti-stalking protections are primarily application-based and require explicit installation and permissions, limiting their effectiveness against passive tracking.

*Takeaway.* Despite different protocol formats and safety triggers, all ecosystems must emit repeated BLE advertisements to remain discoverable by nearby helpers. Ecosystem-specific rotation policies can reduce *identifier-layer* linkability, but they do not eliminate *RF-layer* persistence. These divergences expose distinct attack surfaces, especially for adversaries that exploit fast identifier changes and duty-cycling, which motivates a solution like AirCatch that remains effective under aggressive rotation and stealthy transmission schedules.

# B CFO features from Ubertooth

*Hamming-weight CFO features from Ubertooth.* When capturing with Ubertooth-class devices, we do not have access to full-rate complex IQ samples; instead, the receiver exposes a per-byte frequency estimate, `FREQEST`, which can be interpreted as a coarse CFO proxy sampled at the byte timescale. Let $f_b \in \mathbb{Z}$ denote the signed `FREQEST` value associated with byte $b$ of a packet (in device-specific LSB units), and let $y_b \in \{0,1\}^8$ denote the corresponding




(a) Apple Find My advertisement structure (AirTag-class).

(b) Google Find My Device advertisement structure.

(c) Samsung Find Network advertisement structure (SmartTag2).

(d) Tile advertisement structure.

**Figure 8: Packet structure for Find My networks across tracker ecosystems.**

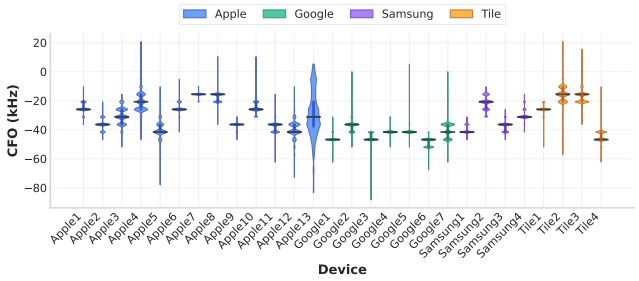



**Figure 9: Ubertooth CFO proxy stability. Per-device distributions of the packet-level FREQEST-derived CFO proxy, showing tight within-device concentration and clear inter-device separation.**

payload byte. We construct *hamming-weight–conditioned CFO* features by binning these CFO proxies by the byte Hamming weight $w_b = \mathrm{HW}(y_b) = \sum_{k=0}^{7} y_{b,k} \in \{0, \ldots, 8\}$ and computing class means:

$$\widehat{\Delta f}_h^{(W)} = \frac{1}{|\mathcal{B}_h|} \sum_{b \in \mathcal{B}_h} f_b, \qquad \mathcal{B}_h = \{\, b \mid \mathrm{HW}(y_b) = h \,\}. \tag{23}$$

Intuitively, conditioning on Hamming weight acts as a *content-aware stratification* of the receiver's frequency proxy: bytes with different Hamming weights induce different transition densities and symbol trajectories under GFSK (after whitening and modulation shaping), which interact with transmitter/receiver filtering memory and discriminator bias. Therefore, while $f_b$ is a coarse statistic, the vector $\{\widehat{\Delta f}_h^{(W)}\}$ captures systematic, hardware-dependent asymmetries that are averaged out by a single global CFO mean. In our implementation, we use a small set of representative bins (e.g., $h \in \{0, 2, 4, 8\}$) through existing USB-exposed fields to avoid protocol changes.

We condition CFO estimates on the Hamming weight of transmitted bytes, yielding the 5-tuple feature:

$$\mathbf{f}_{\mathrm{HW}} = \left[\widehat{\Delta f}_{\mathrm{packet}},\ \widehat{\Delta f}_{w=0},\ \widehat{\Delta f}_{w=2},\ \widehat{\Delta f}_{w=4},\ \widehat{\Delta f}_{w=8}\right], \tag{24}$$

where $\widehat{\Delta f}_{w=k}$ denotes the mean frequency offset accumulated over all bytes of Hamming weight $k$ within the packet.

*Stability and discriminability of features.* The Ubertooth-based analysis in Fig. 9 shows that even a coarse, byte-timescale CFO proxy yields a *stable* per-device signature: for most devices, the per-packet CFO distribution is tightly concentrated (narrow violins), while the centers of these distributions differ substantially across

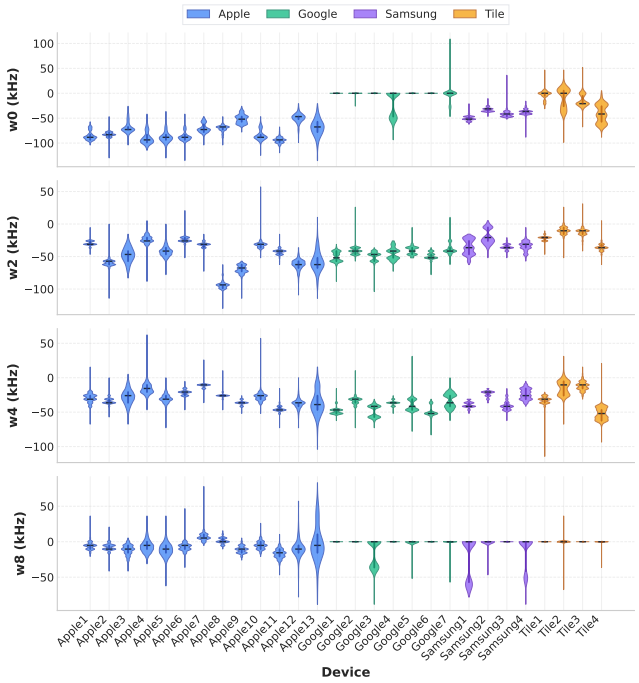



**Figure 10: Hamming-weight–conditioned CFO features (Ubertooth). Conditioned CFO means across representative Hamming-weight bins (e.g., $w \in \{0, 2, 4, 8\}$) provide additional device-specific evidence beyond a single global CFO proxy.**

devices and tag families (separation on the order of *tens of kHz*), enabling reliable inter-device discrimination despite measurement noise. The Hamming-weight–conditioned features in Fig. 10 further strengthen distinctiveness: within each device, the conditioned CFOs remain consistent across packets (small intra-device spread), yet the *relative pattern* across bins (e.g., $w0$ vs. $w2$ vs. $w4$ vs. $w8$) is markedly device-specific, providing additional dimensions to separate devices whose global CFO partially overlaps. Together, these plots empirically validate that our modulation-aware CFO features are both *persistent* and *discriminating*, making them well-suited as an effective physical-layer fingerprint even under receiver constraints.