Passive Multi-Target GUTI Identification via Visual-RF Correlation in LTE Networks

Byeongdo Hong[†], Gunwoo Yoon
The Affiliated Institute of ETRI
byeongdo@nsr.re.kr, gwyoon@nsr.re.kr

***Abstract*—LTE networks employ Globally Unique Temporary Identifiers (GUTIs) to shield subscribers from permanent International Mobile Subscriber Identity (IMSI) exposure, yet we show that these identifiers can be resolved and linked to specific devices through passive observation without prior knowledge of targets. We correlate time-stamped visual observations of device use with over-the-air control-plane messages captured using commodity Software-Defined Radios (SDRs). A Finite-State-Machine (FSM) algorithm processes the synchronized streams to resolve each device's GUTI within the camera's Field of View (FoV), requiring as few as three observed user interactions when the corresponding control-plane messages are captured.**

Field experiments across multiple commercial Long-Term Evolution (LTE) networks validate multi-target resolution: In some deployments, we observed GUTIs persisting for up to 33 days, with reassignment behaviors that were often linkable. Once linked, these long-lived identifiers enable hierarchical location tracking—from cell to paging-area scale—by passively monitoring paging and Radio Resource Control (RRC) messages. Unlike active IMSI catchers or prior GUTI attacks that require pre-existing identifiers (e.g., phone numbers) and active probing, our approach is listen-only and scales to multiple devices within view.

I. Introduction

Cellular networks are core infrastructure for modern society, relied upon daily by billions of people. As of late 2023, approximately 5.6 billion individuals subscribe to mobile services [1]. To mitigate exposure of the permanent International Mobile Subscriber Identity (IMSI), Long-Term Evolution (LTE) introduces the Globally Unique Temporary Identifier (GUTI) to provide identifier privacy by design. This paper demonstrates that, in commercial networks, identifier-privacy protections can be weakened by a non-transmitting adversary who correlates visual observations of device use with the timing of control-plane messages.

IMSI-exposure threats (e.g., IMSI-catchers [2]) have long been reported, including suspected malicious use at courts, airports, and public demonstrations, and they typically affect non-designated populations. By contrast, most GUTI-focused attacks to date are targeted and active: they assume prior contact information such as a victim's phone number or messaging account and induce paging via calls, SMS, or app notifications [3], [4], [5]. These approaches can be effective for a single designated target but face limitations in scaling to multiple targets in parallel and in avoiding detection.

We propose a passive, non-transmitting method to identify the GUTIs of multiple devices in parallel. A camera records timestamps of user interactions, while a Software-Defined Radio (SDR) passively collects co-temporal control-plane events. Our Finite-State-Machine (FSM)-based time-correlation algorithm aligns the two streams and, after observing at least three interactions per device, resolves and verifies the device–GUTI mapping within the camera's Field of View (FoV). The method does not replace active targeted techniques; it operates under different assumptions (no prior contact information, passive collection, multi-target scope).

Across multiple commercial LTE networks in two countries, none of the operators we measured regularly updated GUTIs; in some cases, GUTIs persisted for up to 33 days, and reallocations followed regular operational patterns. Once a person–GUTI association is established, disambiguating paging and Radio Resource Control (RRC) messages suffices to enable hierarchical location inference from the cell level to paging-area scale. Compared with IMSI-catchers and active GUTI attacks, our approach is fully passive, reduces the detection surface, and can simultaneously cover all users within the camera's FoV.

Our contributions are as follows:

- **Passive Multi-Target Threat Model.** We formalize a new threat model that identifies GUTIs for multiple devices via camera–Radio Frequency (RF) time correlation within the FoV.
- **FSM-Based Identification Algorithm.** We present an algorithm that establishes and verifies device–GUTI mappings using as few as three observed user interactions and is robust to missing control-plane messages.
- **Persistence and Reallocation Measurements.** We empirically measure GUTI lifetimes and reallocation patterns in commercial networks, quantifying differences between standard recommendations and operational practice.
- **Real-World Validation.** We demonstrate simultaneous multi-target identification in live LTE environments and analyze operational characteristics of a fully passive approach.

[†] Corresponding author: waybrightly@gmail.com

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.242487
www.ndss-symposium.org

Implications and Disclosure. These findings indicate that, in environments already covered by surveillance cameras (e.g., public or site security Closed-Circuit Television (CCTV)), an adversary equipped with off-the-shelf SDR receivers can turn short visual observations into longer-term RF-only tracking of specific devices. Multi-week GUTI persistence reduces the observation needed for long-term tracking, creating concrete privacy risks for ordinary users. We followed responsible disclosure with GSMA[1] Security team, who acknowledged the issue and agreed to disseminate it to member organizations.

The remainder of this paper is organized as follows. Section II provides background and related work. Section III describes our measurement environments, data-collection methodology, and experimental setup. Section IV analyzes GUTI allocation, persistence, and reallocation patterns in commercial LTE networks. Sections V and VI present our GUTI identification framework and FSM-based identification algorithm. Section VII reports real-world multi-target experiments, and Section VIII evaluates robustness, parameter sensitivity, and scalability. Section IX discusses limitations, mitigations, and 5G feasibility; Section X concludes, and Section XI details ethical considerations.

II. Background and Related Work

This section provides background and related work on GUTI identification and contextualizes our methodology.

A. Background

Cellular networks utilize a specific architecture and protocol stack to support user mobility and data services [6]. Cellular networks deploy a range of broadcast channels, and many control-plane and system messages on these channels are transmitted unencrypted.

1) Identifiers in Cellular Networks: Permanent identifiers broadcast over channels pose privacy risks. Therefore, mobile devices in cellular networks use temporary identifiers for communication. In LTE, subscribers are identified at the Non-Access Stratum (NAS) layer by the GUTI, and its shortened form, the SAE-Temporary Mobile Subscriber Identity (S-TMSI), is used in RRC procedures and paging. A GUTI encodes the operator and Mobility Management Entity (MME) context together with the MME Temporary Mobile Subscriber Identity (M-TMSI). The S-TMSI includes the MME Code and the lower 32 bits (M-TMSI), which distinguish individual users within an MME. Because the remaining GUTI fields are effectively fixed for a given operator and region, learning the S-TMSI is tantamount to learning a linkable GUTI for that subscriber. Accordingly, throughout this paper we use "GUTI identification" to refer to the acquisition of the S-TMSI.

In the Medium Access Control (MAC) [7] layer, devices receive a unique Cell-Radio Network Temporary Identifier (C-RNTI, hereafter RNTI) on a per-cell basis. This identifier remains valid only while the device is connected to the base station. On the device's subsequent connection, a new RNTI is allocated.

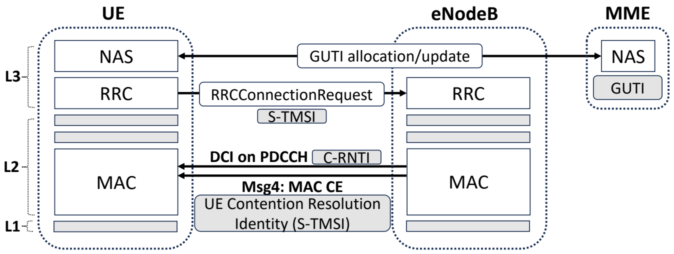


Fig. 1: Protocol path of LTE identifiers

As shown in Figure 1, the Physical Downlink Control Channel (PDCCH) carries Downlink Control Information (DCI) [8], [9] with the RNTI, which encodes only downlink/uplink scheduling information; it does not carry any GUTIs. The subscriber's temporary identity appears at the NAS layer as the GUTI, whose shortened form S-TMSI is included in the uplink `RRCConnectionRequest`. When contention resolution uses an identity-based procedure, the eNodeB echoes this S-TMSI as the UE Contention Resolution Identity in a MAC control element in downlink Msg4, which is typically multiplexed with the `RRCConnectionSetup` in the same MAC Protocol Data Unit (PDU). Our GUTI identification procedure first correlates RNTIs observed in DCI with the MAC-layer Contention Resolution Identity (CRI) to recover the S-TMSI, and then uses this to obtain the corresponding GUTI, yielding the RNTI↔S-TMSI/GUTI mapping.

2) Broadcast Messages in Cellular Networks: In broadcast channels, the Paging Control Channel (PCCH) is used for paging, prompting devices in the `RRC IDLE` state to re-establish an RRC connection when downlink activity is pending. The RRC `Paging` message carried on PCCH includes User Equipment (UE) identities such as the S-TMSI to specify which devices should respond.

The PDCCH schedules data for mobile devices by carrying DCI, which contains the RNTI along with resource allocation, Modulation and Coding Scheme (MCS), and Physical Resource Block (PRB) information.

RRC [10], [11] is a protocol layer to manage radio resources in cellular networks. To conserve battery, most mobile devices remain in a state where the RRC connection is released, known as `RRC IDLE`. When a UE in this state is paged for downlink data, it performs a random access and RRC connection establishment procedure, transitioning to the `RRC CONNECTED` state. The base station completes the connection setup by sending an `RRCConnectionSetup` message.

B. Related Work

Several studies have obtained identifiers or personal information by analyzing broadcast messages in cellular networks. **IMSI Exposure and Catcher Detection.** Early defenses against IMSI catchers focused on UE-side tools such as SnoopSnitch and AIMSICD, which inspect baseband logs

[1] GSMA (GSM Association) is a global association representing mobile network operators and related industries. GSMA members include mobile operators, device manufacturers, software companies, equipment providers, and internet companies from across the globe.

for suspicious cipher-suite changes or identity requests [12], [13]. However, they only observe a single handset and are prone to false positives whenever operators legitimately reconfigure broadcast parameters. To widen the observation coverage, systems like SeaGlass and Crocodile Hunter deploy geo-distributed radio sensors and flag rogue cells based on anomalies in network topology or signal power [14], [15]. Legacy IMSI-catcher threats in LTE networks are well documented [2], [16], and Tucker *et al.* still report such activity in LTE control-plane messages in the wild [17]. However, both IMSI catchers and their detectors rely on active transmissions or observable RF changes, whereas our multi-target GUTI identification operates in a fully passive, listen-only mode with no RF footprint.

GUTI Identification. Prior work has shown that temporary identifiers in 2G and LTE often persist for long periods or follow predictable reassignment patterns [3], [4], [5]. These paging-based methods analyze the paging channel but require prior knowledge of the victim (e.g., phone number or messenger ID) to trigger paging via calls or messages. In contrast, our identification algorithm uses only visual observation of device usage and passively captured RF bursts, eliminating the need for phone numbers, active engagement, or protocol manipulation.

Attacks using GUTI. Many studies build on tools that decode broadcast information in cellular networks [18], [19], [20], [21], [22], [23], [24], [25], [26]. Among these, several studies propose attacks that leverage GUTI. Some cause signaling overshadowing and injection based on a GUTI [27], [28]; others aim to track locations more precisely than the cell level [29], [30]; and some can launch DoS attacks on targeted users [31], [32]. Other studies map GUTIs to RNTIs to acquire DCI and fingerprint websites [33], [34] or videos [35], [36], [37]. If obtaining GUTIs is difficult in practice, these attacks become significantly harder to carry out.

While these studies demonstrate various GUTI-based attacks, they assume GUTI acquisition as given. Our work addresses this fundamental gap by providing a practical, passive method to identify GUTIs without requiring phone numbers, active transmission, or protocol manipulation.

III. Measurement and Data-Collection Methodology

We first describe the measurement environments, equipment, and datasets used throughout this paper. Unless otherwise noted, all experiments were conducted on commercial LTE networks using receive-only (passive) observation; we did not transmit any RF signals nor interact with the operators' infrastructure. All experimental participants were members of the research team, and no third-party or non-consenting users were included as observation targets.

a) Network environments: Measurements were collected from four commercial LTE Mobile Network Operators (MNOs) in two countries (Country A and Country B), denoted MNO-I–MNO-III for Country A and MNO-IV for Country B. For the identifier distribution study (Section IV), we selected indoor locations and urban-area cafes as environments where mobile data traffic is naturally generated. All experiments were performed by passively receiving RF signals or by observing only our own devices.

b) Equipment and logging pipeline: The data-collection pipeline consists of three components: (i) an SDR-based downlink sniffer, (ii) device-side diagnostic logging, and (iii) timestamped visual recording.

SDR Capture. We connect a Universal Software Radio Peripheral (USRP) B210 to a laptop and run the open-source SRS AirScope tool to capture LTE downlink control and broadcast channels. SRS AirScope decodes PDCCH messages (DCI) and RRC signaling, producing time-stamped records of RNTIs, scheduling decisions, and control-plane events. Over the entire measurement campaign, the PDCCH processing rate ranged from 73 % to 100 %, and our evaluation explicitly accounts for this range.

Device-level Ground Truth. To obtain ground-truth mappings between UEs, RNTIs, and GUTIs, we use the commercial diagnostic tool XCAL [38] on selected test devices. XCAL receives control-plane data directly from the mobile device's diagnostic port and logs RNTIs, GUTIs, and various radio parameters. In addition, when possible, we read debugging screens provided by Samsung Galaxy devices to verify GUTI allocation patterns (e.g., when GUTIs are re-assigned or reused). These logs are used only for offline validation and are assumed to be inaccessible to the attacker model.

Visual activity monitoring. User interactions with the test devices are recorded using a Galaxy S25 Ultra smartphone running a timestamp camera application. This application overlays millisecond-precision wall-clock timestamps on each frame. In the 3-UE and 10-UE experiments, the camera is positioned so that multiple devices on a table are visible at once, capturing coarse gestures such as screen taps, swipes, and device pickups. The attacker does not require access to detailed screen content; the core attack relies on timing of visual "in-use" events, with optional filters using coarse service cues only when available.

c) Time synchronization: During experiments, the laptop connected to the SDR is synchronized with a Network Time Protocol (NTP) server, while the timestamp camera and the observed devices obtain their time from the cellular network. The offset between the NTP server time and the cellular network time remained within 1 s, and the time thresholds used in our observation framework easily cover this bound.

d) Datasets: Using the above setup, we collected three types of datasets that are used throughout the paper.

(i) *D1: network-level control-plane traces* are SDR captures from MNO-I and MNO-II in Country A. For Section IV, we use multi-cell D1 traces containing DCI and RRC messages to characterize anonymized active RNTIs and GUTIs in operational networks. A long-term subset of D1, logged over 11 days (3 h 31 min of active monitoring) and comprising 4,096 unique GUTIs, is used in Section VIII to evaluate GUTI similarity and to estimate the probability that two different devices exhibit colliding data-service times.

(ii) *D2: GUTI allocation-pattern traces* consist of periodic screenshots of Samsung Galaxy debug screens, which expose the current physical cell, RF parameters, and the GUTI assigned to the device. We capture these screens intermittently and analyze them to study how GUTIs are re-assigned or recycled in practice.

(iii) *D3: visual-RF correlation experiments* consist of 12 controlled experiments (E1–E12) in which 3 or 10 test devices are placed within the camera's FoV while the SDR captures downlink signals from the serving cell. These traces combine timestamped visual triggers, RF bursts, and device-level ground truth derived from XCAL, and are used both to illustrate service-specific DCI patterns and to evaluate our GUTI identification framework (Section VII).

IV. Measurement of Identifier Characteristics

LTE networks assign multiple identifiers to each subscriber in order to deliver data services. Our work investigates the risks that arise from the mapping between those identifiers and higher-layer information. In this section we analyze the real-world distribution of identifiers and their associated metadata, showing that even short-lived identifiers can ultimately pose long-term privacy risks. While our empirical analysis focuses on LTE, prior measurement studies of 5G temporary identifiers in other regions have reported similarly long-lived behavior, suggesting that operator practices can yield persistent "temporary" identifiers across both LTE and 5G deployments [39], [40].

A. Identifier Distribution

Using a USRP B210 [41] SDR and the SRS AirScope software [24], we captured `RRCConnectionSetup` messages and downlink DCIs at two geographically separated sites and for two different operators (MNO-I and MNO-II) in Country A. From each `RRCConnectionSetup` we counted the number of unique GUTIs to approximate the population of attached devices, whereas the per-minute count of active RNTIs in the live DCI stream served as a proxy for concurrent data users. Figure 2 plots the time-of-day results: depending on location and operator, the median number of active RNTIs ranged from roughly 126 to 267 per minute, while the number of visible GUTIs varied between 206 and 536. These figures provide a first-order estimate of both the total device population registered to a single cell and the subset that is actively sending or receiving payload data at any given moment.

B. Service-Specific Traffic Patterns

Real-time DCI messages reveal, on a sub-frame basis, the exact amount of uplink (UL) and downlink (DL) resources scheduled for each UE. Figure 3 visualizes these allocations as a heat-map of UL + DL byte counts, grouped by service type and device; darker tiles indicate larger bursts. Due to measurement limitations, some allocations may be missed (e.g., one Galaxy S24 voice call), yet several stable patterns stand out:

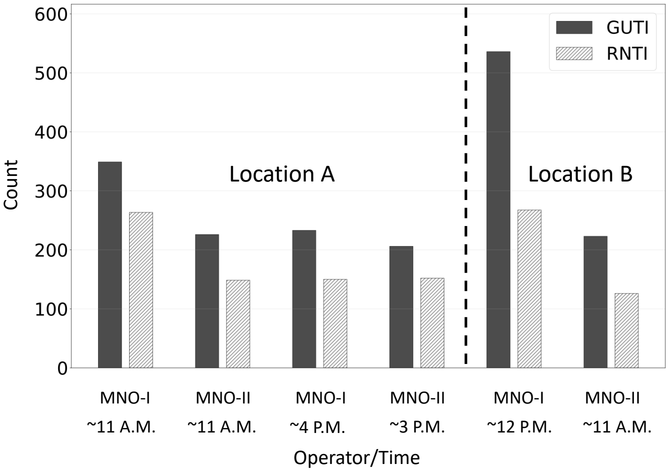


Fig. 2: Statistics of GUTI and RNTI. The tilde (∼) indicates approximate measurement times within a 1-hour window

- **Voice Calls.** Calls maintain consistently low data rates (UL+DL $<$ 5 kB/s), appearing as sustained low-intensity allocations that reflect the codec's constant bit rate.
- **Video Streaming.** Bursts occur intermittently with varying intensity based on buffering strategies and content bitrate. In some sessions the base station releases the RRC context and assigns a fresh RNTI mid-stream (iPhone 7: 0x725f $\rightarrow$ 0xcf48 $\rightarrow$ 0xed56); in others, fewer RNTI changes occur (Galaxy S22 Ultra: 0x0044).
- **Web Browsing.** Burst patterns show high variability: brief, intense spikes (up to 100+ kB/s) corresponding to page loads, followed by idle periods. Data volume varies dramatically based on content type.

These service-specific signatures remain visible in the control plane and can be correlated with GUTI/RNTI transitions, underscoring the viability of our identification and profiling capabilities.

C. GUTI Persistence and Predictability

GUTIs in LTE networks should ideally be updated frequently to ensure user anonymity. However, our measurements indicate that these identifiers can remain unchanged for long periods or follow predictable patterns when reallocated. We also observed that a non-negligible portion of their bit space remains fixed, allowing attackers to infer the next identifier after reallocation.

1) Unchanging GUTI: An attack remains valid as long as the GUTI mapped to a particular subscriber remains unchanged. In our experiments on two mobile operators in Country A, GUTIs persisted for up to 33 days without reallocation under normal network conditions (i.e., without powering off or using airplane mode). In Country B (MNO-IV), we also observed that `Service Request` procedures did not trigger GUTI updates, although our limited measurement window did not allow us to quantify long-term GUTI lifetimes in that network.

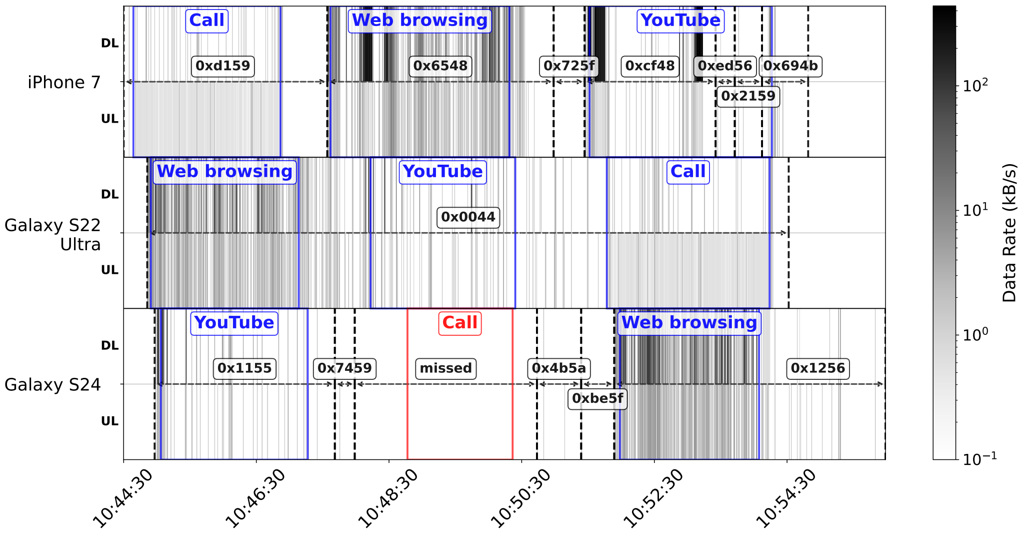


Fig. 3: DCI-derived traffic patterns for different mobile services

2) Predictable Reassignments: Our measurements confirm that some operators do not fully randomize GUTI updates. For example, we have observed scenarios where the first 12 or 16 bits remained identical, or bits 9–25 (17 bits) remained static upon each reallocation, effectively creating a predictable "prefix." A prior global-scale measurement study spanning 11 countries [5] showed that if an attacker can infer the sequence in which a network reassigns GUTIs, they can reliably map each reallocation to the same subscriber and thus keep identifying the victim's new GUTI. Our results show that certain MNOs in Country A continue to exhibit this behavior, indicating that GUTI updates are either infrequent or follow patterns that are linkable in practice.

3) Implications: Consequently, once a GUTI is revealed, an adversary can keep tracking the user even if the operator performs a nominal reallocation, since the attacker can infer the next GUTI from the partial prefix carried over. Prior work further shows that exposed GUTIs can be used to recover the victim's IMSI [28], [42], amplifying the impact of long-lived and linkable temporary identifiers. This finding aligns with the broader observation that GUTI reallocation does not necessarily improve privacy if operators reuse bits in a predictable manner.

V. GUTI Identification Framework

The fact that any visually observable user can be exposed to long-term privacy leakage without prior knowledge represents a serious threat. This section delineates how such exposure becomes feasible by detailing our method for extracting the victims' GUTIs.

A. Overview

We introduce a framework that extracts a device's GUTI without prior knowledge of the target using only line-of-sight observations of its mobile activity. Figure 4 gives a high-level overview. A timestamp-synchronized camera records coarse mobile-usage events (e.g., when the user appears to use data, place a voice call, or browse the web) without requiring access to screen content or exact service labels, and a laptop sniffer simultaneously captures over-the-air control-plane traffic. By aligning video frames with sniffer logs, we fuse each visible action with the corresponding data-burst pattern and identifier metadata. The resulting traces are fed into an FSM algorithm that outputs the GUTI for every device in view. Once a GUTI is bound to a specific user, the identifier is no longer anonymous and can be exploited by the privacy attacks surveyed in Section II-B.

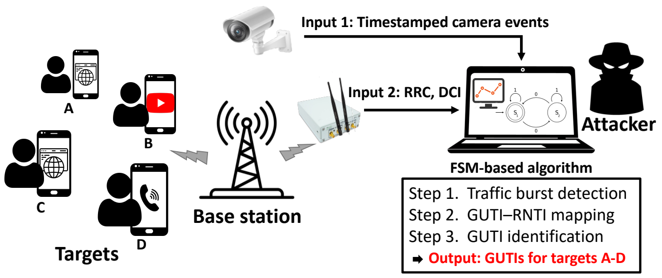

Input 1: Timestamped camera events
A
B
C
D
Targets
Base station
Input 2: RRC, DCI
Attacker
FSM-based algorithm
Step 1. Traffic burst detection
Step 2. GUTI–RNTI mapping
Step 3. GUTI identification
Output: GUTIs for targets A-D

Fig. 4: Overview of GUTI identification framework

B. Requirements

To passively extract a victim's GUTI, the attacker needs two capabilities:

R1. **Timestamped Observation of Mobile Activity.** The attacker must know *when* the target triggers a packet-data event (e.g., making a VoLTE call, tapping a web link, or starting a YouTube video). A simple CCTV, smartphone camera, or external Universal Serial Bus (USB) webcam that embeds millisecond-resolution timestamps suffices. No knowledge of the exact application is required—only a binary indication of "phone in use." When the service type exhibits distinct patterns—such as continuous flows for voice calls, periodic bursts for video streaming, or sporadic traffic for web browsing—our algorithm leverages these characteristics to accelerate GUTI extraction and improve filtering accuracy.

R2. **LTE Radio-Interface Sniffing.** The attacker needs an SDR setup capable of (i) decoding `RRCConnectionSetup` messages and (ii) demodulating downlink DCIs on the PDCCH. This is readily achievable with open-source stacks such as srsRAN [19] or OpenAirInterface [20], or with commercial tools like SRS AirScope [24].

Clock alignment. The camera host and the radio sniffer need only a loose synchronization: an offset within ±0.5 s is adequate for the 4 s matching window used in our experiment (Section VII). In our experiments, the laptop clock was synchronized via NTP, whereas the smartphone camera relied on the phone's network-provided time, resulting in a residual skew below 200 ms.

Minimal Observation Time. At least *two* `RRCConnectionSetup` messages from the same device must be captured. The first setup yields a set of candidate GUTIs, while the second allows us to resolve the GUTI uniquely (Section VI). In our FSM, a third observation is typically used as a verification step before accepting the mapping. Hence the total observation time depends on the user's inter-activity interval.

C. Visual-RF Correlation Principle

Our GUTI extraction framework leverages a fundamental observation: user-initiated mobile activities generate predictable RF burst patterns. When a user interacts with their device (e.g., clicking a link, starting a video, sending a message), the application triggers network traffic that manifests as data bursts in the cellular interface. By correlating visual observations of these interactions with RF measurements, we can isolate the target device's traffic among multiple active UEs in the cell.

This visual–RF correlation is essential because:

- **Timing Precision**: User actions provide precise timestamps for expected data bursts (on the order of 100–500 ms).
- **Ground Truth**: Visual confirmation ensures that we track the correct device throughout the experiment.
- **Service Differentiation (optional)**: Different activities produce distinct traffic patterns—sporadic bursts (web browsing), regular chunks (video streaming), or continuous streams (voice calls). In our framework, such patterns are used only as an optional accelerator; the FSM core requires only binary activity (in use vs. idle).

D. Threat Model

We consider an adversary with the following capabilities:

1) Visual observation of device-usage events for users within the field of view of one or more cameras
2) Passive cellular signal capture using commodity SDR equipment located in the same cell sector
3) Correlation of observed activities with RF transmissions

This threat model encompasses realistic attack scenarios:

- **Public Surveillance**: Observing device usage patterns in cafes, airports, or public transportation
- **Behavioral Monitoring**: Detecting when users actively interact with their devices (typing, tapping, holding to ear)
- **Proximity Attacks**: Identifying data service activation without screen visibility

This threat model primarily captures adversaries that already operate or can covertly access fixed cameras in public or semi-public spaces (e.g., building CCTV operators, site security, or state-level actors with access to video infrastructure). We do not claim that opportunistic attackers can deploy this attack everywhere; instead, our focus is on camera-monitored environments where visual observation is already taking place.

While an attacker could also deliberately bring their own camera and SDR to specific locations and opportunistically select targets (e.g., at a protest or event), such attacks still require setting up and maintaining line-of-sight visual monitoring and are therefore more constrained than purely RF-only schemes such as IMSI catchers.

The visual requirement is key: the attacker must maintain line-of-sight to the devices long enough to observe a small number of coarse usage events (screen taps, device pickups, holding the phone to the ear). Importantly, our attack does not rely on screen contents or fine-grained biometrics—binary "in-use vs. idle" information is sufficient—but the attack remains limited to users who fall within the cameras' field of view and the coverage of the sniffer's serving cell.

VI. GUTI Identification Algorithm

We propose an FSM-based GUTI identification algorithm that systematically identifies and extracts GUTI values by correlating physical-layer observations with application-layer activities. The algorithm employs a state-based approach to ensure robust tracking while preventing infinite loops and handling edge cases in real-world cellular environments.

Our FSM consumes two asynchronous streams: timestamped device-usage events from the camera and RF bursts captured by the SDR that carry both scheduling information (DCI with RNTIs) and the MAC CRI. Each new pair of camera events and RF bursts constrains which GUTIs could have generated the observed activity, progressively shrinking the candidate set. Once a single GUTI remains consistent with all past observations, the FSM regards this GUTI as the current best candidate and enters a final verification phase that checks whether subsequent bursts continue to match this hypothesis over time.

At a high level, the FSM operates in four stages (Algorithm 1). (i) `INIT` performs basic setup, preparing the camera and RF input streams and initializing the candidate set. (ii) `SCAN` monitors the RF trace and, for each observed camera event, searches within a configurable temporal window for DCI-derived burst traffic and associated RNTIs; this window is chosen large enough to cover the clock offset between the camera (cellular time) and the SDR host (NTP time). (iii) `COLLECT` and `FILTER` then extract GUTIs from the identified bursts and prune inconsistent candidates using camera-derived usage events, iterating until the candidate set becomes empty or only a single GUTI remains. (iv) `VERIFY` finally checks whether this remaining candidate consistently explains future bursts within the verification timeout window τ_{ver}. Appendix A provides the formal state-space definition and extended transition details.

Formal Data Burst Definition. We use a binary predicate Burst(RNTI, T) to indicate whether an RNTI exhibits significant DL/UL activity around time T. We define a data burst for an RNTI at time T as follows:

Definition 1 (Data Burst)**.** *A data burst occurs at time T for an RNTI when the cumulative data within a time window exceeds a threshold:*

$$\textit{Burst}(\textit{RNTI}, T) = \begin{cases} 1, & \textit{if} \sum\limits_{t \in [T - t_{th}, T + t_{th}]} (DL_t + UL_t) \geq d_{th}, \\ 0, & \textit{otherwise}. \end{cases} \tag{1}$$

Here, t_{th} denotes the time threshold window, d_{th} represents the data threshold in bytes, and DL_t, UL_t correspond to downlink and uplink data at time t, respectively.

VII. Real-World Experiments

Having presented our GUTI identification framework, we demonstrate its real-world feasibility and privacy implications through field experiments. Despite GUTI's design goal of preventing user tracking, we show that our framework successfully extracts these temporary identifiers and enables long-term user tracking in commercial networks.

A. Experimental Setup

Our experimental setup consists of four main components: target devices, RF capture equipment, visual activity monitoring, and time synchronization.

Target Devices. For single- and three-UE experiments (E1–E6), we used four smartphones from different manufacturers and models: iPhone 7 (A10 Fusion), Galaxy S20 5G (Snapdragon 865), Galaxy S22 Ultra (Snapdragon 8 Gen 1), and Galaxy S24 (Exynos 2400). For the 10-UE experiments (E7–E8), we used ten commercial smartphones (one iOS and nine Samsung Galaxy models; a full list is given in Section VII-B3). For cross-country experiments in Country B (E9–E12), we targeted Galaxy S23 Ultra and Galaxy S24. To validate RNTI allocation and GUTI assignment at the device level, we employed XCAL [38], a commercial diagnostic tool that extracts control-plane messages from the baseband processor. While alternative tools exist (e.g., QXDM [43], QCSuper [44], Network Signal Guru [45], MobileInsight [46], and SCAT [47]), we used XCAL in our experiments due to its stable support for both iPhone and Samsung Galaxy devices in our setup. For the iPhone, we specifically chose the iPhone 7 as newer models restrict baseband diagnostics; this model allows diagnostic profile installation [48], enabling RNTI and GUTI monitoring through XCAL. For Samsung Galaxy devices, we additionally leveraged the vendor-provided engineering/debug screen, which exposes serving-cell and NAS identifiers, to directly confirm the GUTI values shown to the user.

Algorithm 1 FSM-based GUTI Identification

```
Require: Timestamped camera events $C$, RF signal trace $R$, maximum
    iterations $N_{\text{iter}}$, verification timeout $\tau_{\text{ver}}$
Ensure: Target GUTI $g^*$ or FAILURE
state ← INIT
$G \leftarrow \emptyset$                              ▷ set of candidate GUTIs
iter ← 0
$g^* \leftarrow \perp$                                ▷ current best hypothesis (if any)
while state $\notin$ {DONE, FAIL_SAFE} do
    if state = INIT then                              ▷ Phase 0: timeline alignment
        AlignTimelines($C, R$)
        state ← SCAN
    else if state = SCAN then                         ▷ Phase 1: wait for bursts
        $B \leftarrow$ DetectBursts($R$)
        if $|B| > 0$ then
            state ← COLLECT
        end if
    else if state = COLLECT then                      ▷ Phase 2: extract raw GUTIs
        for all $b \in B$ do
            $g \leftarrow$ ExtractGUTI($b$)
            if $g \neq \perp$ then
                $G \leftarrow G \cup \{g\}$
            end if
        end for
        state ← FILTER
    else if state = FILTER then                       ▷ Phase 3: prune via camera events
        $G \leftarrow$ ApplyFilters($G, C$)
        iter ← iter + 1
        if iter > $N_{\text{iter}}$ then              ▷ prevent non-terminating loops
            state ← FAIL_SAFE
        else if $|G| = 1$ then                        ▷ single consistent candidate found
            $g^* \leftarrow$ the unique element of $G$
            state ← VERIFY
        else if $|G| = 0$ then                        ▷ no candidate survived filtering
            $G \leftarrow \emptyset$, $B \leftarrow \emptyset$
            state ← SCAN
        else                                          ▷ multiple candidates remain; wait for more events
            state ← SCAN
        end if
    else if state = VERIFY then                       ▷ Phase 4: verify final hypothesis
        $t_{\text{start}} \leftarrow$ currentTime
        repeat
            burstOK ← Verified($g^*$)
        until burstOK $\vee$ currentTime $- t_{\text{start}} > \tau_{\text{ver}}$
        if burstOK then
            state ← DONE
        else
            state ← SCAN
            $G \leftarrow \emptyset$, $B \leftarrow \emptyset$, $g^* \leftarrow \perp$
        end if
    end if
end while
if state = DONE then
    return $g^*$
else
    LogError($G$, iter)                               ▷ state must be FAIL_SAFE here
    return FAILURE
end if
```

RF Signal Capture. We deploy one SDR sniffer per monitored downlink E-UTRA Absolute Radio Frequency Channel Number (EARFCN). In E1–E6 and E9–E12, a single

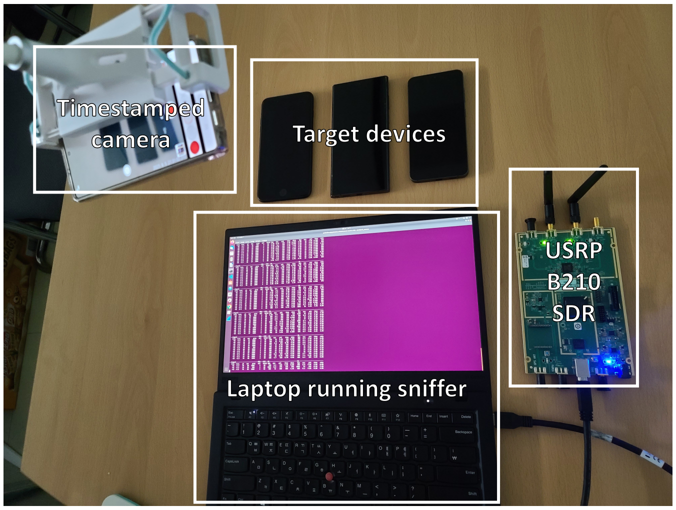


Fig. 5: Experimental setup for GUTI identification

USRP B210 [41] paired with SRS AirScope [24] sufficed because UEs camped on one EARFCN. In E7–E8, we monitored two downlink EARFCNs, so we used two USRP B210 instances running SRS AirScope, each tuned to one EARFCN. Across our experiments, per-sniffer processing rates ranged from 73 % to 100 %, depending on RF conditions and host load. For the primary sniffer, we attached the SDR to a laptop with an Intel i7-8650U CPU; when deploying an additional sniffer in E7–E8, we used a mobile workstation with an Intel Xeon E-2176M CPU.

Visual Activity Monitoring. We recorded device interactions in all experiments using a Galaxy S25 Ultra smartphone running a timestamp camera application [49] that overlays millisecond-precision timestamps on the video feed. By placing the target UEs in a single field of view, we could simultaneously monitor up to ten UEs (E7–E8). This visual capture serves two purposes: (1) providing trigger points for expected data bursts when users interact with their devices, and (2) establishing ground truth for validating extracted GUTIs against actual device usage. While our experimental setup captures detailed service information for comprehensive evaluation, we emphasize that the framework requires only binary knowledge of data activity (active/inactive) for successful GUTI extraction. The detailed service classification enhances performance but is not a prerequisite.

Time Synchronization. The SDR laptop was synchronized to an NTP server, while the recording smartphone relied on cellular network time, ensuring a consistent time reference across our experimental setup. While minor logging delays may occur due to processing latency, these fall within our time-window parameter (t_{th}) and do not affect the algorithm's performance.

Figure 5 depicts our complete experimental environment incorporating all components described above.

FSM Parameterization. Unless stated otherwise, all experiments execute the FSM with the parameter set in Table I. The data-burst threshold is fixed to $d_{th} = 10\,\text{kB}$ and the time window to $t_{th} = 2\,\text{s}$, meaning that a burst is recognized once the aggregate DL+UL volume for an RNTI exceeds 10 kB within any 4-second sliding window. Other parameters follow the design choices in Section VI and Appendix A: the continuous-traffic gap $\tau_{\text{gap}} = 300\,\text{ms}$, verification timeout $\tau_{\text{ver}} = 300\,\text{s}$, and the iteration cap $N_{\text{iter}} = 10$ to prevent infinite loops.

TABLE I: FSM runtime parameters used in all experiments

Symbol	Value	Unit	Role
d_{th}	10	kB	data-burst threshold
t_{th}	2	s	time-window half-width
τ_{gap}	300	ms	maximum inter-burst gap (continuous)
τ_{ver}	300	s	VERIFY-state timeout
N_{iter}	10	iteration	FILTER-state loop bound

B. *GUTI Identification Results*

We conducted twelve experimental sessions (E1–E12) across four MNOs in two countries, totaling 37 per-UE GUTI-identification attempts. We achieved 36/37 (97 %) successful extractions and 35/37 (94 %) full verifications overall. The only extraction failure occurred on Galaxy S23 Ultra in E11, and the other verification failure occurred on Galaxy S24 in E5 (see Table II).

1) Single-Device Experiments (E1–E4): Four isolated tests were performed with individual devices, each executing mixed services (voice calls, YouTube, web browsing) sequentially. All experiments achieved successful GUTI extraction and verification despite control-plane packet losses, with only 1–2 missed service instances per session.

Representative case E1 demonstrates the algorithm's robustness: the FSM successfully identified a single GUTI immediately after the first call service ended, with verification completed during the subsequent YouTube session. When shifting the observation starting point to the second service (YouTube), the GUTI was uniquely identified during the web browsing phase, with verification completed at the next web navigation. Notably, even when the sniffer missed the initial call service, the FSM's FILTER state produced an empty candidate set, triggering automatic recovery. The algorithm then resumed tracking from the fifth service (YouTube), successfully extracting the GUTI during the second YouTube session and completing verification during the final web service.

In E4, despite missing the third (YouTube) and fourth (web) services, the algorithm extracted and verified the GUTI from the first web service alone. When shifting the observation point to the call service, the GUTI was still uniquely identified after call termination, with verification completed during the next call despite intermediate missed services.

2) Multi-Device Experiments (E5–E6): To demonstrate universality beyond specific manufacturers or chipsets and validate multi-target tracking capabilities, we conducted multi-device experiments using diverse hardware: an iPhone with Apple silicon, and Samsung devices with both Qualcomm and Exynos chipsets.

TABLE II: Experimental results of GUTI identification

Exp.	Sniffer(s)[‡]	Device(s)	Chipset	Network	Services[†]	# Missed	Extraction	Verification
E1	1 (99 %)	Galaxy S20	Snapdragon 865	MNO-II	C–Y–W–C–Y–W	1 (4)	✓	✓
E2	1 (93 %)	Galaxy S20	Snapdragon 865	MNO-II	W–Y–W–Y–W–Y–C	1 (2)	✓	✓
E3	1 (90 %)	Galaxy S20	Snapdragon 865	MNO-I	Y–W–C–Y–W–C	1 (5)	✓	✓
E4	1 (90 %)	Galaxy S22	Snapdragon 8 Gen1	MNO-I	W–C–Y–W–C	2 (3, 4)	✓	✓
E5	1 (84 %)	iPhone 7	A10 Fusion	MNO-II	Y–W–A–Y–W–A	1 (6)	✓	✓
		Galaxy S22	Snapdragon 8 Gen1		Y–W–C–Y–W–C	3 (1, 2, 5)	✓	✓
		Galaxy S24	Exynos 2400		W–C–Y–Y–W–C	4 (1, 2, 5, 6)	✓	✗[*]
E6	1 (90 %)	iPhone 7	A10 Fusion	MNO-I	C–W–Y–W–C–W–Y	0	✓	✓
		Galaxy S22	Snapdragon 8 Gen1		W–Y–C–W–Y–C	0	✓	✓
		Galaxy S24	Exynos 2400		Y–C–W–Y–C–W	3 (3, 5, 6)	✓	✓
E7	2 (100 %, 100 %)	10 UEs	Various	MNO-II	W (intermittent)[§]	0	✓	✓
E8	2 (100 %, 89 %)	10 UEs	Various	MNO-III	W (intermittent)[§]	6[¶]	✓	✓
E9	1 (78 %)	Galaxy S23 Ultra	Snapdragon 8 Gen2	MNO-IV[‖]	Y–G–W–G–W–Y	0	✓	✓
E10	1 (73 %)	Galaxy S23 Ultra	Snapdragon 8 Gen2	MNO-IV[‖]	Y–G–W–G–W	2 (2, 3)	✓	✓
		Galaxy S24	Exynos 2400		G–W–Y–W–Y	1 (4)	✓	✓
E11	1 (81 %)	Galaxy S23 Ultra	Snapdragon 8 Gen2	MNO-IV[‖]	W–G–G–Y–G–W	5 (2–6)	✗	✗
		Galaxy S24	Exynos 2400		Y–W–Y–W–Y	2 (2, 4)	✓	✓
E12	1 (74 %)	Galaxy S23 Ultra	Snapdragon 8 Gen2	MNO-IV[‖]	Y–W–G–W–G–Y	0	✓	✓
		Galaxy S24	Exynos 2400		W–G–Y–G–G–Y	0	✓	✓

[*] GUTI extracted but verification incomplete due to high packet loss and limited observation window.
[†] Service legend: C=Call, Y=YouTube, W=Web, A=Apple TV, G=Google Meet (audio/video).
[‡] Sniffer(s): "n (p)" or "n (p_1, p_2)" denotes n SDR sniffers with per-sniffer processing rate(s) in percent.
[§] Each UE performed intermittent web browsing; we use data-service presence, not the specific service type.
[‖] MNO-I/II/III are operators in Country A, while MNO-IV is an operator in Country B.
[¶] "# Missed" = 6: one each on iPhone 7, Galaxy S10 5G, S20 5G, Z Fold5, and two on S23 Ultra.

E5 (MNO-II) operated under challenging conditions with approximately 16 % control packet loss (84 % processing rate), resulting in multiple missed services. Nevertheless, the algorithm successfully extracted and verified GUTIs for most observation cases. The worst-case scenario (Galaxy S24 in E5) illustrates a practical limitation: while GUTI extraction succeeded, verification could not be completed due to missing 4 out of 6 services. Despite this, the algorithm correctly narrowed to a single GUTI during consecutive YouTube services separated by a brief idle period. Subsequent services were all missed, preventing target verification. All other devices in the multi-device experiments achieved both successful extraction and verification.

3) Large-Scale and Cross-Country Experiments (E7–E12): To further validate our framework with a larger device set and in additional countries, we conducted 10-UE experiments on Country A's MNO-II (E7) and MNO-III (E8). We used ten commercial smartphones: one iOS device (iPhone 7, A10 Fusion) and nine Samsung Galaxy devices—S10 5G (Exynos 9820), Note10+ 5G (Exynos 9825), S20 5G (Snapdragon 865), S21 5G (Exynos 2100), S22 Ultra (Snapdragon 8 Gen 1), S23 (Snapdragon 8 Gen 2), S23 Ultra (Snapdragon 8 Gen 2), S24 (Exynos 2400), and Z Fold5 (Snapdragon 8 Gen 2). At each measurement location, UEs typically camped on a primary downlink EARFCN, but in practice EARFCN reselection can occur. To avoid missing control-plane messages during such changes, we deployed sniffers not only on the primary EARFCN but also on the additional EARFCN observed in that area. In E7–E8, all UEs intermittently browsed the web for experimental convenience. Across all experiments, we annotate service types to characterize usage patterns, but our GUTI extraction logic primarily relies on the presence and timing of data-triggered control-plane activity, so using web traffic here does not materially limit generality.

In E7, both sniffers achieved a processing rate of 100 %, unlike in other experiments, and thus captured all control-plane messages for every target without loss. Consequently, we were able to extract and verify the GUTI of every device using only three observed data-service events per device. In E8, the two sniffers processed at 100 % and 89 %, respectively. As summarized in Table II, this resulted in one missed message each for the iPhone 7, Galaxy S10 5G, S20 5G, and Z Fold5, and two missed messages for the S23 Ultra, while all other devices experienced no message loss. Despite these losses, our framework still successfully extracted and verified the GUTI for all devices, including those with missed messages, by leveraging additional observations.

Experiments E9–E12 were conducted on operator MNO-IV in Country B, which is distinct from Country A's MNO-I/II/III, and targeted Galaxy S23 Ultra and Galaxy S24 devices. These measurements were performed in a specific localized area where the sniffer processing rates were relatively low (73–81 %). Nevertheless, except for a single failure case on the Galaxy S23 Ultra in E11, we were able to identify the GUTI in all runs. In the E11 failure case, the sniffer missed all subsequent control-plane messages after the first data-service event, ultimately preventing GUTI identification.

4) Analysis of Missed Service Cases: While our experiments demonstrated successful GUTI extraction, sniffer performance remains critical as capturing target devices' RRC and DCI messages is fundamental to the attack. In E1, despite 99 % processing rate, the sniffer missed the target's `RRCConnectionSetup` while successfully capturing all DCI messages. Among single-device experiments, only E4 experienced both DCI and RRC losses for the target, while others only missed RRC messages. Multi-device experiments presented debugging challenges due to limited multi-device debugger connectivity, making it difficult to determine whether missed cases were due to DCI, RRC, or other issues. While higher processing rates generally improve robustness, our results show that average rate alone is not predictive. In Country B, we successfully extracted GUTIs at processing rates as low as 73–81 % (E10/E12) when losses were dispersed, but also observed a failure at 81 % (E11) due to a long blackout immediately after the first data-service event. This indicates that sustained observation can sometimes compensate for moderate loss, yet cannot fully recover from bursty gaps around key trigger events. In summary, lower processing rates can still lead to successful GUTI identification given sufficient observation time, but fast and reliable extraction in practice requires maintaining a high processing rate around trigger bursts.

Overall, our experiments indicate that while distinctive services such as voice calls or web browsing can speed up convergence by producing clearer burst patterns, the core FSM logic primarily relies on the timing of data-triggered control-plane activity rather than precise service labels.

C. Attack Scenarios: Location Tracking

Using an identified GUTI, a passive adversary can perform hierarchical location tracking at three granularities: (i) cell-level presence via `RRCConnectionSetup`, (ii) paging-area-level presence via PCCH paging messages, and (iii) absence from the paging area when the GUTI disappears from both channels. This attack requires only passive monitoring and remains fully exploitable in current LTE deployments.

The feasibility of hierarchical tracking stems from the different scopes of paging and RRC signaling. When a device in `RRC IDLE` receives downlink traffic, the network broadcasts paging messages for the device's GUTI over a paging area that may span multiple cells, whereas `RRCConnectionSetup` occurs only in a single physical cell. By monitoring both PCCH and RRC signaling with a single sniffer, an adversary can therefore infer whether a target is located inside a specific cell, somewhere within the paging area, or outside paging coverage altogether.

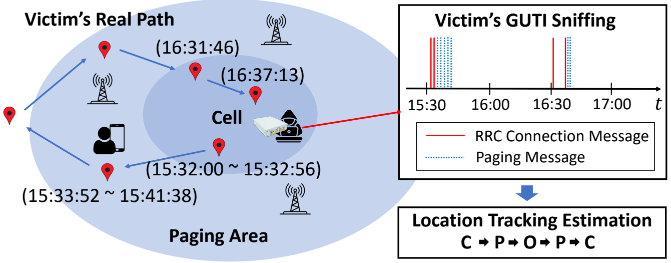


Fig. 6: Experimental results of hierarchical location tracking with a single passive sniffer

Across two operators (MNO-I and MNO-II), we measured the ratio of unique GUTIs observed on PCCH versus RRC to range from 10 to 38, indicating that paging areas are approximately 10–38 times larger than individual cells. In our measurement locations, the physical cell area was approximately 50,000–90,000 m^2, corresponding to paging areas of roughly 0.5–3.6 km^2. These values are consistent with prior studies showing that paging areas can span tens of square kilometers depending on operator policies and service types.

In a representative case study, we tracked a device as it moved between a serving cell, the surrounding paging area, and outside paging coverage using only passive observation (Figure 6). Even when the device was not actively using data services, periodic paging messages enabled coarse-grained localization. In our measurements, paging messages were transmitted on the order of a few times per hour for idle devices, while `RRCConnectionSetup` events allowed rapid confirmation of cell-level presence when data activity occurred.

VIII. Evaluation

This section evaluates our GUTI identification framework using real-world traces and operational measurements. We first estimate how many visually observed device-usage events are needed to uniquely resolve a GUTI, and then empirically measure false-positive and false-negative rates of our burst detector under realistic capture loss. We further study the sensitivity of the FSM thresholds τ_{gap} and τ_{ver}, and finally compare our attack with existing identifier-extraction techniques.

A. Required Number of Observations

The method of extraction fundamentally requires the observation of the target's mobile activity. We analyzed the required number of observations to extract GUTI based on actual PDCCH data. Our validation data from two MNOs in Country A were logged over 11 days, totaling 3 hours, 31 minutes, and 8 seconds. We identified a total of 4,096 unique GUTIs in the dataset. We then analyzed similarities between GUTIs based on the start times of data services throughout this period. For experimental evaluation, we measured the similarity of data start times among random users, not the target.

Definition 2 (GUTI Similarity). *Let two GUTI timelines be*

$$g_i = \{(RNTI_{i_1}, t_{i_1}), \dots, (RNTI_{i_n}, t_{i_n})\},$$
$$g_j = \{(RNTI_{j_1}, t_{j_1}), \dots, (RNTI_{j_k}, t_{j_k})\},$$

where each pair $(RNTI_x, t_x)$ denotes the RNTI that "initiated a data-burst" at time t_x, i.e., Burst$(RNTI_x, t_x) = 1$ as defined in Definition 1. We say that g_i and g_j are "similar" iff

$$n = k \quad and \quad |t_{i_\ell} - t_{j_\ell}| < t_{th} \;\; \forall \ell \in \{1, \ldots, n\}.$$

In words, the two GUTIs expose the same sequence length of burst-triggering RNTIs and their corresponding burst-start times match within a tolerance of t_{th} seconds.

We analyzed all the verification data we recorded and extracted every RNTI mapped to each GUTI, logging the activation time and the amount of data for each RNTI. This allowed us to construct a data-usage timeline for each GUTI. When the time threshold t_{th} is set to 2 seconds, 197 pairs of similar GUTIs were found among 4,096 unique IDs. Notably, all these similar GUTIs had only one RNTI. Not a single pair of GUTIs that mapped to two or more RNTIs exhibited similarity. By increasing the time threshold to 5 seconds, the count of similar GUTIs increased to 485 pairs. Among these, one pair had 2 RNTIs, and another had 3 RNTIs; all remaining 483 pairs consisted of GUTIs mapped to a single RNTI. With a 2-second time threshold, observing two mobile-activity events is typically sufficient to identify a user with high probability in our dataset. With a 5-second threshold, the identification success rate increases to 99.902 % (4,092/4,096).[2]

When a user in `RRC IDLE` mode uses data services, a new RNTI is assigned. This analysis shows that the extraction method can identify a single GUTI when it is possible to identify a user's RNTI twice, that is, when two observations can be made.

B. Error-Rate Analysis (10 kB / 2 s detector)

Theoretical bounds on the performance of any identifier-extraction scheme are elusive because mobile traffic is highly non-stationary: burst inter-arrival times and sizes depend on location, access technology, time of day, and user behavior. Instead of attempting full analytical coverage, we estimate false-positive (FP) and false-negative (FN) rates empirically, using the real-world trace set described in Section VIII-A. All results assume the production detector parameters $d_{th} = 10$ kB and $t_{th} = 2$ s (cf. Section VI).

a) False positives: An FP occurs when the FSM selects and verifies a non-target GUTI g_a although the true target is g_t. For this to happen (i) g_a must survive `FILTER` as a unique candidate and (ii) pass `VERIFY`, which requires at least $n_{\min} = 3$ aligned data bursts. In the 211-minute trace in our experiment, the probability that two unrelated RNTIs produce three bursts within $t_{th} = 2$ s is 0. Even with an exaggerated 5 s window the joint probability remains below $0.1\,\%$. Therefore, even in a worst-case setting—looser $t_{th} = 5$ s and the sniffer simultaneously missing all bursts of g_t—the overall FP risk stays $< 0.1\,\%$.

TABLE III: FN rate versus burst opportunities ($p = 0.8096$)

	3	4	5	6	7	8
$\Pr[E_n]$	0.531	0.834	0.930	0.971	0.986	0.994
FN rate (%)	46.9	16.6	7.0	2.9	1.4	0.6

b) False negatives: An FN is registered when the algorithm fails to verify the actual target. The dominant factor is control-plane loss at the sniffer; environmental conditions (no user traffic, RF shadowing) simply postpone extraction rather than degrading accuracy.

Burst-Capture Model. With the USRP pipeline sustaining $\geq\; 90\,\%$ throughput, the trace shows a per-burst capture probability $p = 0.8096$ (i.e., $19.04\,\%$ of bursts missed). Let each burst opportunity be an independent Bernoulli trial with success probability $p = 0.8096$ ($q = 1 - p = 0.1904$). For a sequence of n opportunities, define

$$E_n = \Big(\sum_{i=1}^{n} s_i \geq 3\Big) \;\wedge\; \Big(\exists j \; (s_j = s_{j+1} = 1)\Big),$$

i.e., *at least three captured bursts in total and at least one occurrence of two successive captures*. The success probability is then

$$\Pr[E_n] = 1 - \underbrace{\sum_{k=0}^{2} \binom{n}{k} p^k q^{n-k}}_{\text{succ.} \leq 2} - \underbrace{\sum_{k=3}^{\lfloor \frac{n+1}{2} \rfloor} \binom{n-k+1}{k} p^k q^{n-k}}_{\text{succ.} \geq 3 \text{ \& no ``11''}}.$$

With five opportunities the FN rate already drops to $7\,\%$, and for $n \geq 6$ it falls below $3\,\%$, demonstrating that three captured bursts provide a practical balance between accuracy and latency.

Table III shows that once the sniffer observes five or more burst opportunities from the handset, the FN rate drops below $7\,\%$; at eight bursts it is already under $1\,\%$. That rate could be reduced further if the sniffer's processing rate increases.

C. Parameter Sensitivity of FSM Thresholds (τ_{gap} and τ_{ver})

We evaluate the sensitivity of the FSM to its parameters τ_{gap} and τ_{ver} by varying them around their default values (Table I) using real-world traces. Unless otherwise noted, all experiments use the default parameter set.

1) Sensitivity of τ_{gap} : The continuous-service filter is not a core requirement of our FSM-based GUTI identification algorithm; the framework can identify a target GUTI even without classifying traffic as continuous or non-continuous. However, when a clearly continuous service is observed within a short interval (e.g., a VoLTE call), this filter provides a useful accelerator by pruning candidates that exhibit long silent gaps.

We analyze this behavior using the notion of no-burst intervals. The formal definition of a no-burst interval is given in Appendix B. Intuitively, a data burst for an RNTI occurs at time T if the cumulative DL/UL data within $[T - t_{th},\, T + t_{th}]$ exceeds d_{th}, and a no-burst interval corresponds to a time span of length at least $2t_{th}$ in which no such detection window contains a burst. The continuous-service filter classifies a flow

[2] Out of the total 4,096 GUTIs, the number of GUTIs, excluding the pair mapped to 2 and 3 RNTIs, is 4,092.

TABLE IV: No-burst intervals for YouTube traffic

Operator	10th perc.	Median	90th perc.
MNO-I	4.64 s	9.22 s	9.98 s
MNO-II	5.43 s	9.47 s	11.46 s
MNO-IV	4.46 s	7.46 s	10.57 s

TABLE V: Verification success rate as a function of the verification timeout τ_{ver} (replayed traces)

τ_{ver} (s)	Success rate (%)
50	0.0
100	70.2
200	89.1
300	94.5

as continuous only if its longest inter-burst gap $L_{\max}$ satisfies $L_{\max} \leq 2t_{th} + \tau_{\text{gap}}$.

With $t_{th} = 2\,\text{s}$ and $\tau_{\text{gap}} = 300\,\text{ms}$, this threshold becomes $2t_{th} + \tau_{\text{gap}} = 4.3\,\text{s}$. Using DCI-derived traffic traces, we observe that YouTube-like non-continuous flows consistently exhibit at least one no-burst interval longer than 4.3 s, whereas continuous services (VoLTE, Google Meet) do not. Figure 7 illustrates this separation, and Table IV summarizes the distribution of no-burst interval lengths for YouTube traffic across three operators. Even the 10th percentile of YouTube no-burst intervals exceeds the threshold in all cases, confirming that $\tau_{\text{gap}} = 300\,\text{ms}$ robustly separates continuous and non-continuous traffic in our measurements. Decreasing τ_{gap} toward zero does not change classification outcomes, while substantially increasing it risks misclassifying non-continuous flows as continuous.

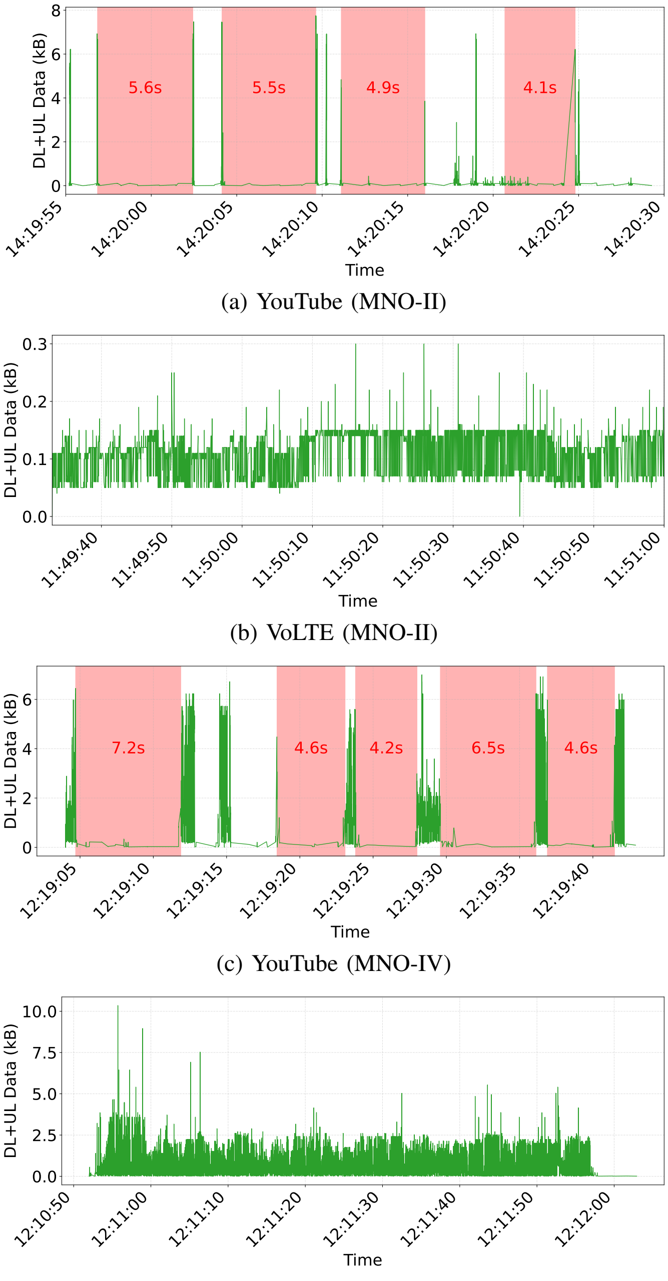
2) Sensitivity of τ_{ver} : The parameter τ_{ver} bounds the duration of the `VERIFY` state after a candidate GUTI–UE mapping has been established. This timeout limits buffer growth and ensures that incorrect mappings are eventually discarded if no confirming activity is observed.

Table V reports the verification success rate obtained by replaying our experimental traces with different timeout values. Very short timeouts are ineffective: at $\tau_{\text{ver}} = 50\,\text{s}$, no mapping is verified because inter-activity intervals in our traces exceed the timeout. As τ_{ver} increases, the success rate improves monotonically and reaches 94.5 % at $\tau_{\text{ver}} = 300\,\text{s}$, matching the outcome of our original experiments without enforced timeouts. This indicates that once τ_{ver} is sufficiently large, further increases primarily affect buffer occupancy rather than the correctness of identification. In practice, moderately generous verification timeouts (on the order of several minutes) provide a good balance between robustness and resource usage.

D. Comparison with Existing Attacks

Table VI compares our FSM-based GUTI identification framework with existing identifier-extraction techniques. Our approach combines passive operation with cell-level coverage, offering stealth advantages while avoiding user-side disruption.

The primary limitation—requiring line-of-sight for camera correlation—is a trade-off for avoiding over-the-air transmission, which reduces the RF footprint compared to active approaches. Unlike IMSI catchers that must transmit rogue signals, our passive approach relies solely on listen-only RF reception and external visual cues, without transmitting or modifying any cellular traffic, making it technically less conspicuous than active attacks.



(a) YouTube (MNO-II)

(b) VoLTE (MNO-II)

(c) YouTube (MNO-IV)

(d) Google Meet (MNO-IV)

Fig. 7: DCI-derived traffic patterns for YouTube, VoLTE, and Google Meet services; the y-axis shows scheduled DL+UL data per 1 ms slot (kB)

IX. Discussion

This section discusses the practical requirements for mounting our attack, key factors that constrain its deployment,

TABLE VI: Identifier-extraction attacks

Attack	Pass.	Prior?	LoS?	Scale	Detect.	Impact
IMSI (BTS) [2]	✗	✗	✗	Cell[1]	High	Service drop
IMSI (Shadow) [27], [28]	✗	✗	✗	Cell	Med.	None
Phone–GUTI [3], [4], [5]	✗	✓	✗	Single	Low	Phone rings
Ours	✓	✗	✓	**Cell**[2]	**V. low**	**None**

[1] Uplink-based; limited by SDR transmit power (typically tens of meters).
[2] Limited by camera line-of-sight; covers only visible UEs in the cell area.
Pass. = Passive?, Prior? = Requires prior identifier?, LoS? = Requires line of sight?, Detect. = Detection likelihood.

discrepancies between security standards and real-world implementations, the feasibility of extending our approach to 5G, and potential mitigation strategies. Taken together, these points clarify where our GUTI identification framework is most applicable in practice and how operators can reduce the associated privacy risks.

A. Practical Attack Requirements

While our experiments employed detailed activity monitoring for thorough evaluation, practical attacks require significantly less—an adversary need only detect when targets actively use their devices. For example, an adversary could simply observe when a user looks at their phone or holds it to their ear.

Multi-EARFCN Deployments. In our experiments, UEs typically camped on a single primary downlink EARFCN, so one sniffer per monitored EARFCN was sufficient. However, in real deployments neighboring cells or overlapping coverage can expose multiple candidate EARFCNs, and inter-frequency handovers may move a device between them near cell borders. In E7–E8 we therefore monitored two downlink EARFCNs using two B210–SRS AirScope instances, each tuned to one carrier, and merged their logs offline before running the FSM. An adversary with access to multiple SDRs or a wideband receiver could apply the same strategy to avoid losing control-plane messages during inter-frequency mobility while still operating passively.

B. Limitations

While our FSM-based GUTI identification approach demonstrates high success rates, several limitations constrain its applicability and generalizability.

Line-of-Sight Requirement. Our method fundamentally requires visual observation of target devices to correlate RF signals with user activities. This constraint limits deployment to scenarios where the adversary can position a camera with clear visibility of targets, excluding environments such as obscured locations, or scenarios where targets are not directly observable.

Single-Cell Coverage. Unlike active IMSI catchers that can actively pull devices into their coverage, our passive approach is limited to the coverage area of a single cell sector where the sniffer is deployed. Targets moving between cells require repositioning of equipment or multiple synchronized collection points.

SDR Performance Constraints. We rely on commercial SDR equipment (USRP B210) that may miss control-plane messages under heavy traffic conditions. Our experiments showed processing rates of 73–100 %, with lower rates often correlating with increased false negatives.

Deployment Variability. Our experiments focused on three MNOs in Country A and one MNO in Country B, all in urban environments. Other regions may use different configurations and traffic loads (e.g., peak vs. off-peak hours), which can affect identification success rates and generalizability.

Network-Specific Implementations. Despite 3GPP standardization, operators implement different GUTI management strategies. Some may use shorter rotation intervals, randomized allocation, or trigger reassignment on specific events, potentially limiting our method's effectiveness.

Multi-Target Tracking. In our experiments, we successfully tracked up to 10 devices simultaneously within a single camera view (E7–E8), and our FSM operates on all visible RNTIs in a receive-all, filter-targets fashion. Scaling beyond tens of devices, however, would require more sophisticated and fully automated video analysis to detect and separate device-usage events for many users in parallel. In particular, reliably tracking hundreds of targets in dense crowds appears limited by the visual factors (per-device event detection, occlusion handling, and identity maintenance) rather than by RF processing complexity, and we leave such large-crowd, fully automated visual tracking as future work.

Experimental Validation Scope. All experiments used researchers' own devices, preventing validation against diverse real-world user behaviors and device types. Moreover, our test networks did not employ strong GUTI-randomization defenses during the measurement period, so we could not validate our framework under operators that aggressively rotate identifiers.

Comparison with Active Attacks. Unlike active IMSI catchers that guarantee target connection, our passive approach cannot force GUTI exposure—targets must naturally generate traffic. Silent or idle devices remain undetectable, and users can evade tracking by disabling data services or entering airplane mode.

These limitations collectively suggest that while GUTI identification remains a practical privacy threat, practical deployment faces significant operational hurdles that may limit its attractiveness compared to active attacks in certain scenarios. Future work should address these constraints while maintaining the passive nature that provides our method's primary advantage.

C. Discrepancy between Standards and Implementation

Cellular networks, often described as *walled gardens* [50], frequently display a gap between established standards and their actual implementation. This gap arises from several factors: the complexity of network architecture, the difficulty for users and MNOs to recognize security issues, and a tendency to prioritize performance over security. The implementation of cellular networks varies by region and carrier, making it challenging to maintain consistent security measures. Moreover, passive attacks, such as location tracking, often go unnoticed, further complicating the enforcement of security. As networks evolve and grow more complex, there is an increasing focus on operational performance at the expense of security enhancements. While safe standards are critical, ensuring strict compliance with these standards is equally essential to protect user privacy.

D. 5G Feasibility

Our work focuses on LTE, which remains the predominant deployment globally, but the same class of risk may also arise in 5G. Recent 5G security updates [51] strengthen protection of permanent identifiers and make classical IMSI-catcher style attacks harder, but they do not directly eliminate exposure of temporary identifiers. In 5G, data scheduling over PDCCH still relies on RNTI-based addressing, and commercial and research analyzers already demonstrate the ability to recover 5G downlink DCIs. For example, tools such as WaveJudge [25] and 5GSniffer [52] can decode DL-DCIs, and more recent tools like NR-Scope [53] reportedly add RRC message decoding, potentially exposing contention-resolution identities and 5G-GUTI assignments. At the same time, NR introduces important differences: only part of the 5G S-TMSI is used in contention resolution, and the mapping between temporary identifiers and control-plane messages differs from LTE.

We do not claim an end-to-end 5G attack pipeline, nor did we validate our framework on 5G standalone deployments. Instead, we treat 5G applicability as a feasibility question pending further verification: existing tools suggest that recovering 5G control information is becoming practical in some environments, but empirical evaluation of 5G-GUTI extraction and tracking is left as future work.

E. Countermeasures and Quantitative Evaluation

Hong *et al.* [5] proposed a cryptographically secure pseudorandom number generator-based GUTI reassignment mechanism for LTE [54], [55], [56], which led the 3rd Generation Partnership Project (3GPP) to update the standard [57] in 2022 (version 17.0.0) so that an unpredictable identifier is assigned on every `Service Request` [58], [59]. When a UE uses non-voice services, frequent RRC reconnections generate regular `Service Requests`, limiting GUTI persistence and hindering extraction. However, an attacker can suppress `Service Requests`—for example, using the scheduling attack of Oh *et al.* [30] to keep the RRC connection active—and still obtain the RNTI and recover the GUTI via reverse identity mapping, enabling long-term tracking until the UE powers off.

Beyond theoretical solutions such as PDCCH encryption, which require protocol-level changes and long deployment cycles, we consider two practical near-term mitigations. The first, aperiodic GUTI renewal, can be realized via core-network policy updates. The second, lightweight deception using decoy identifiers, would require more invasive changes to scheduling and identifier management and is therefore more speculative.

a) Aperiodic GUTI renewal: Beyond periodic GUTI rotation tied to mobility or service requests, we consider a network-initiated renewal policy that enforces a maximum lifetime for each GUTI. Unlike periodic or event-driven schemes, renewal is triggered solely by the expiration of this lifetime and is not synchronized with UE behavior or other network events. Once a time threshold T has elapsed since the last assignment, the core triggers a GUTI Reallocation Command, independent of the UE's RRC state (idle or connected). This requires only a policy change in the core network rather than a protocol modification.

To cover a range of plausible operator choices, we consider three example thresholds, $T \in \{300, 600, 1800\}\,\mathrm{s}$ (5, 10, and 30 minutes), ranging from relatively aggressive to more conservative renewal intervals. In our trace-driven what-if analysis, these values shorten GUTI lifetimes from the multi-day to multi-week range observed in our measurements to minute-scale durations. In particular, $T = 300\,\mathrm{s}$ reflects a conservative threshold for substantially reducing the effectiveness of our attack: Section VII shows that our framework can identify a device's GUTI whenever it is observed for roughly five minutes without a long service gap, so enforcing renewal on this timescale either prevents successful identification or, when renewal occurs shortly after identification, renders the identified GUTI too short-lived to be useful for tracking. A GUTI reallocation procedure every five minutes incurs only modest NAS signaling overhead in our setting, but operators can adjust T to match their own policies and overhead budgets—for example, varying the renewal interval by time of day or adopting longer thresholds. Even thresholds of 30 minutes or more still substantially strengthen privacy compared to the current practice of multi-day persistence.

b) Lightweight deception: Another complementary direction is to inject fake control-plane activity to confuse the attacker. Our FSM-based attack gradually narrows the candidate set of GUTIs by intersecting the sets observed at multiple visual triggers; thus, the attack fails if at least two plausible GUTIs remain consistent across all observed triggers. A simple way to increase this ambiguity is to maintain a pool of decoy RNTI–GUTI pairs and, for a fraction $X\,\%$ of real GUTIs that initiate RRC connections, create corresponding decoy twins whose DCI scheduling patterns mimic those of the real targets. When $X = 100\,\%$, our identification procedure can no longer compress the candidate set to a single GUTI by design. In our trace-driven replay, we instantiate $X \in \{70, 50, 30\}$ and observe that the GUTI identification success rate reported in Section VII (94.5 %) drops to 28.3 %, 47.2 %, and 66.1 %, respectively. If the sniffer occasionally fails to decode the target's DCI while still decoding the decoy's DCI, the effective success rate drops even further. That said, such lightweight deception requires careful design of decoy identifier selection and tracking policies, and its protection may be weaker in practice than the aperiodic GUTI renewal mechanism discussed above.

X. Concluding Remarks

This work demonstrates that LTE's GUTI-based privacy protection remains vulnerable to passive adversaries. By correlating timestamped visual cues with broadcast control-plane traffic, an attacker can extract and track users' temporary identifiers without prior knowledge or active interaction. In our field experiments, the FSM-based identification algorithm achieved high success rates, and we observed that, in some networks, GUTIs can persist for weeks and exhibit reassignment patterns that are linkable in practice. Although 3GPP specifications require that GUTIs be reassigned unpredictably, our measurements indicate that operators do not always apply these mechanisms strictly in practice, often favoring operational efficiency. This underscores the need for stricter GUTI policies and for protocol designs that anticipate multi-channel correlation attacks.

XI. Ethical Considerations

We carefully considered the ethical implications of our GUTI identification algorithm. All experiments were conducted exclusively on our own test devices, with researchers serving as the only subjects. No third-party or non-consenting users were monitored, and no GUTIs were extracted from individuals outside the research team. The validation dataset in Section VIII used only anonymized aggregate statistics from passive reception, with no individual identification attempts. Geographic information has been anonymized (Country A, Country B) to prevent operator identification.

We acknowledge that the techniques described in this paper could potentially be misused for unauthorized surveillance purposes. To address this concern, we reported our findings to the GSMA, and the GSMA shared them with its members so they could design and deploy mitigations before this paper's publication.

To prevent unauthorized surveillance applications, we provided specific countermeasures that network operators can implement, including enhanced GUTI rotation schemes and stronger NAS-level protection for subscriber identifiers. We strongly condemn any attempt to use these techniques for unauthorized surveillance or privacy violations, and understanding these vulnerabilities is essential for developing more robust privacy protections in cellular networks.

Our research adheres to the Menlo Report [60] principles for ethical ICT research: Respect for Persons through exclusive use of consenting researchers as subjects; Beneficence by ensuring the security benefits outweigh potential risks; Justice by not targeting any specific population; and Respect for Law and Public Interest through responsible disclosure and providing defensive countermeasures.

References

[1] GSMA Intelligence, "The Mobile Economy 2024," Available: https://www.gsmaintelligence.com/research/the-mobile-economy-2024 (accessed Nov. 12, 2025).

[2] D. Strobel, "IMSI Catcher," *Chair for Communication Security, Ruhr-Universität Bochum*, vol. 14, 2007.

[3] D. F. Kune, J. Koelndorfer, N. Hopper, and Y. Kim, "Location Leaks on the GSM Air Interface," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2012.

[4] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.-P. Seifert, "Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems," *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2016.

[5] B. Hong, S. Bae, and Y. Kim, "GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier." in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2018.

[6] 3GPP. TS 36.401 v18.1.0, "E-UTRAN; Architecture description," 2024.

[7] 3GPP. TS 36.321, "Medium Access Control (MAC) protocol specification," 2019.

[8] 3GPP. TS 36.101, "User Equipment (UE) radio transmission and reception," 2017.

[9] 3GPP. TS 36.212, "Multiplexing and channel coding," 2018.

[10] 3GPP. TS 38.331, "Radio Resource Control (RRC); Protocol specification," 2023.

[11] 3GPP. TS 36.331, "LTE RRC Protocol specification," 2017.

[12] SnoopSnitch, 2022, Available: https://github.com/srlabs/snoopsnitch (accessed Nov. 11, 2025).

[13] SecUpwN, "Android IMSI-Catcher Detector," Available: https://cellularprivacy.github.io/Android-IMSI-Catcher-Detector/ (accessed Nov. 11, 2025).

[14] P. Ney, I. Smith, G. Cadamuro, and T. Kohno, "SeaGlass: Enabling City–Wide IMSI-Catcher Detection," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 3, pp. 26–46, 2017.

[15] "Crocodile Hunter: Cellular Network Security Observatory," https://www.eff.org/pages/crocodile-hunter, Electronic Frontier Foundation, 2021, software, (accessed Aug. 6, 2025).

[16] F. van den Broek, R. Verdult, and J. de Ruiter, "Defeating IMSI Catchers," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015, pp. 340–351.

[17] T. Tucker, N. Bennett, M. Kotuliak, S. Erni, S. Capkun, K. Butler, and P. Traynor, "Detecting IMSI-Catchers by Characterizing Identity Exposing Messages in Cellular Traffic," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2025.

[18] "OsmocomBB Project," Available: http://osmocom.org (accessed Nov. 11, 2025).

[19] "srsRAN_4G," Available: https://github.com/srsLTE/srsLTE (accessed Nov. 11, 2025).

[20] "OpenAirInterface," Available: https://github.com/openairinterface (accessed Nov. 11, 2025).

[21] S. Kumar, E. Hamed, D. Katabi, and L. Erran Li, "LTE Radio Analytics Made Easy and Accessible," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 211–222, 2014.

[22] N. Bui and J. Widmer, "OWL: A Reliable Online Watcher for LTE Control Channel Measurements," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, 2016, pp. 25–30.

[23] R. Falkenberg and C. Wietfeld, "FALCON: An Accurate Real-Time Monitor for Client-Based Mobile Network Data Analytics," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–7.

[24] "SRS AirScope," Available: https://srs.io/tag/airscope/ (accessed Nov. 11, 2025).

[25] KEYSIGHT, "SJ001A WaveJudge Wireless Analyzer Toolset," Available: https://www.keysight.com/us/en/product/SJ001A/wavejudge-5000.html (accessed Nov. 11, 2025).

[26] T. D. Hoang, C. Park, M. Son, T. Oh, S. Bae, J. Ahn, B. Oh, and Y. Kim, "LTESniffer: An Open-source LTE Downlink/Uplink Eavesdropper," in *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2023, pp. 43–48.

[27] H. Yang, S. Bae, M. Son, H. Kim, S. M. Kim, and Y. Kim, "Hiding in Plain Signal: Physical Signal Overshadowing Attack on LTE," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 55–72.

[28] S. Erni, M. Kotuliak, P. Leu, M. Roeschlin, and S. Capkun, "AdaptOver: Adaptive Overshadowing Attacks in Cellular Networks," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 743–755.

[29] N. Lakshmanan, N. Budhdev, M. S. Kang, M. C. Chan, and J. Han, "A Stealthy Location Identification Attack Exploiting Carrier Aggregation

in Cellular Networks," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3899–3916.

[30] T. Oh, S. Bae, J. Ahn, Y. Lee, T. D. Hoang, M. S. Kang, N. O. Tippenhauer, and Y. Kim, "Enabling Physical Localization of Uncooperative Cellular Devices," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 1530–1544.

[31] H. Kim, J. Lee, E. Lee, and Y. Kim, "Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1153–1168.

[32] S. R. Hussain, M. Echeverria, I. Karim, O. Chowdhury, and E. Bertino, "5GReasoner: A Property-Directed Security and Privacy Analysis Framework for 5G Cellular Network Protocol," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 669–684.

[33] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Breaking LTE on Layer Two," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1121–1136.

[34] K. Kohls, D. Rupprecht, T. Holz, and C. Pöpper, "Lost Traffic Encryption: Fingerprinting LTE/4G Traffic on Layer Two," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019, pp. 249–260.

[35] S. Bae, M. Son, D. Kim, C. Park, J. Lee, S. Son, and Y. Kim, "Watching the Watchers: Practical Video Identification Attack in LTE Networks," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1307–1324.

[36] N. Lakshmanan, A. Bentaleb, B. Choi, R. Zimmermann, J. Han, and M. S. Kang, "On Privacy Risks of Watching YouTube over Cellular Networks with Carrier Aggregation," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 1, pp. 1–22, 2022.

[37] J. Baek, P. K. D. Soundrapandian, S. Kyung, R. Wang, Y. Shoshitaishvili, A. Doupé, and G.-J. Ahn, "Targeted Privacy Attacks by Fingerprinting Mobile Apps in LTE Radio Layer," in *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2023, pp. 261–273.

[38] "XCAL," Available: https://www.accuver.com/sub/products/view.php?idx=6 (accessed Nov. 11, 2025).

[39] S. Nie, Y. Zhang, T. Wan, H. Duan, and S. Li, "Measuring the Deployment of 5G Security Enhancement," in *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2022, pp. 169–174.

[40] O. Lasierra, G. Garcia-Aviles, E. Municio, A. Skarmeta, and X. Costa-Pérez, "European 5G Security in the Wild: Reality versus Expectations," *arXiv preprint arXiv:2305.08635*, 2023.

[41] "USRP B210," Available: https://www.ettus.com/product/details/UB210-KIT (accessed Nov. 11, 2025).

[42] M. Kotuliak, S. Erni, P. Leu, M. Röschlin, and S. Čapkun, "LTrack: Stealthy Tracking of Mobile Phones in LTE," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1291–1306.

[43] "QxDM Professional," Available: https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/80-n9471-1_d_qxdm_professional_tool_quick_start.pdf (accessed Nov. 11, 2025).

[44] "QCSuper Tool," Available: https://github.com/P1sec/QCSuper (accessed Nov. 11, 2025).

[45] "Network Signal Guru," Available: https://www.qtrun.com/eng/nsg/ (accessed Nov. 11, 2025).

[46] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, and T. Wang, "MobileInsight: Extracting and Analyzing Cellular Network Information on Smartphones." in *Proceedings of the ACM Annual International Conference on Mobile Computing & Networking (MobiCom)*, 2016.

[47] "Signaling Collection and Analysis Tool," Available: https://github.com/fgsect/scat (accessed Nov. 11, 2025).

[48] Apple Inc., "Baseband and Telephony Logging Profile for iOS/iPadOS," Available: https://developer.apple.com/bug-reporting/profiles-and-logs/ (accessed Nov. 11, 2025).

[49] Bian Di, "Timestamp Camera," Available: https://play.google.com/store/apps/details?id=com.jeyluta.timestampcamerafree (accessed Nov. 11, 2025).

[50] B. Hong, S. Park, H. Kim, D. Kim, H. Hong, H. Choi, J.-P. Seifert, S.-J. Lee, and Y. Kim, "Peeking over the Cellular Walled Gardens-A Method for Closed Network Diagnosis," *IEEE Transactions on Mobile Computing*, vol. 17, no. 10, pp. 2366–2380, 2018.

[51] 3GPP. TS 33.501, "Security architecture and procedures for 5G System," 2023.

[52] "5GSniffer," Available: https://github.com/spritelab/5GSniffer (accessed Nov. 11, 2025).

[53] "NR-Scope: A 5G Standalone Cellular Network Telemetry Tool," Available: https://github.com/PrincetonUniversity/NR-Scope (accessed Nov. 11, 2025).

[54] NIST, SP, "800-90a revision 1," *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, 2015.

[55] W. Kan, "Analysis of Underlying Assumptions in NIST DRBGs." *IACR Cryptology ePrint Archive*, vol. 2007, p. 345, 2007.

[56] K. Q. Ye, "The Notorious PRG: Formal Verification of the HMAC-DRBG Pseudorandom Number Generator," Princeton, NJ, Apr. 2016, department of Computer Science. Available: https://www.cs.cmu.edu/~kqy/resources/thesis.pdf (accessed Nov. 11, 2025).

[57] 3GPP. TS 33.401, "3GPP System Architecture Evolution (SAE); Security architecture," 2023.

[58] 3GPP, "S3-220061, CR 0702, Align GUTI allocation to best practices of unpredictable identifier generation." Available: https://portal.3gpp.org/ChangeRequests.aspx?q=1&specnumber=33.401 (accessed Nov. 11, 2025).

[59] ——, "TSG-SA3 Meeting #106-e, S3-220075, GUTI allocation discussion paper," Available: https://portal.3gpp.org/ChangeRequests.aspx?q=1&specnumber=33.401 (accessed Nov. 11, 2025).

[60] "The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research," US Department of Homeland Security, 2012.

Appendix A
Detailed FSM Algorithm

A. State Space Definition

Definition 3 (FSM State Space)**.** *The FSM consists of seven distinct states:*

$$S = \{S_0, S_1, S_2, S_3, S_4, S_5, S_6\} \tag{2}$$

where:

- S_0 *(*`INIT`*): System initialization*
- S_1 *(*`SCAN`*): Cell scanning and RNTI monitoring*
- S_2 *(*`COLLECT`*): Initial GUTI candidate collection*
- S_3 *(*`FILTER`*): Candidate reduction*
- S_4 *(*`VERIFY`*): Final verification*
- S_5 *(*`FAIL_SAFE`*): Exception handling*
- S_6 *(*`DONE`*): Successful completion*

B. FSM State Specifications

1) State S_0 (INIT): The initialization state performs system setup with the following operations:

- Calibrate the time offset between camera and sniffer, and set the temporal matching tolerance
- Initialize data structures: $G \leftarrow \emptyset$, iter $\leftarrow 0$
- Calibrate RF frontend

Exit transition: $S_0 \xrightarrow{\text{init\_done}} S_1$

2) State S_1 (SCAN): The scanning state monitors cellular activities:

- Monitor `RRCConnectionSetup` messages
- Track DCI allocations on PDCCH
- Build RNTI $\leftrightarrow$ GUTI mapping cache

Exit transition: $S_1 \xrightarrow{\text{burst\_detected}} S_2$

3) State S_2 (COLLECT): For each RNTI with detected burst, extract corresponding GUTI from `RRC Setup` messages and update candidate set: $G \leftarrow G \cup \{\text{extracted GUTIs}\}$. Exit transition: $S_2 \xrightarrow{\text{candidates\_created}} S_3$

TABLE VII: FSM state transition specifications

Current	Event	Condition	Next
INIT	init_done	sync_success	SCAN
SCAN	burst_detected	burst_count > 0	COLLECT
COLLECT	candidates_created	$\|G\| > 0$	FILTER
FILTER	empty or multiple	$\|G\| \neq 1$	SCAN
FILTER	single	$\|G\| = 1$	VERIFY
FILTER	exceeded	iter $> N_{\text{iter}}$	FAIL_SAFE
VERIFY	success	BurstMatch	DONE
VERIFY	timeout	$t > \tau_{\text{ver}}$	SCAN

4) State S_3 (FILTER): Apply burst pattern and service-specific filters to reduce candidate set.
Exit transitions:

$$S_3 \xrightarrow{|G|\neq 1} S_1 \quad \text{(empty set or multiple candidates)} \tag{3}$$

$$S_3 \xrightarrow{|G|=1} S_4 \quad \text{(single candidate)} \tag{4}$$

$$S_3 \xrightarrow{\text{iter}>N_{\text{iter}}} S_5 \quad \text{(limit exceeded)} \tag{5}$$

5) State S_4 (VERIFY): By the end of `FILTER`, the candidate set has been reduced to a single GUTI, denoted g^*. VERIFY therefore (i) continuously tracks the RNTI mapped to g^* and (ii) upon the target's next observed service-usage event, examines whether that RNTI results in a Data Burst.
Verification predicate.

$$\text{Verified}(g^*) = \begin{cases} 1, & \text{if BurstMatch}(g^*) \\ 0, & \text{otherwise.} \end{cases}$$

* BurstMatch(g^*) – at least one data burst associated with the RNTI that g^* maps to is detected while the target device is visibly using a service (camera trigger window).

If the predicate is satisfied within a verification timeout τ_{ver} (default 300 s), the FSM marks the GUTI as verified; otherwise the algorithm returns to `SCAN` and the candidate set is reset.

C. State Transition Table

The complete state transition logic of the FSM is summarized in Table VII. Each transition is triggered by a specific event and guarded by a condition that must be satisfied. The FSM ensures deterministic behavior by defining mutually exclusive conditions for all transitions from each state. Notably, the FILTER state has three possible transitions based on the cardinality of the candidate set G, while terminal states (DONE and FAIL_SAFE) have no outgoing transitions.

The transition from FILTER to SCAN when $|G| = 0$ implements automatic recovery, allowing the algorithm to restart data collection without full reinitialization. This design choice significantly improves resilience against temporary signal loss or false positive bursts.

D. Continuous Service Filtering

For services requiring persistent connections (e.g., Voice over IP (VoIP)), a candidate is kept if its maximum inter-burst gap $\max_i(\Delta t_i)$ is below the threshold

TABLE VIII: Filtering parameters and default values

Symbol	Default	Unit	Meaning / Tuning basis
τ_{gap}	300	ms	maximum value of inter-burst gap
Δt_i	—	ms	gap between burst i and $i{+}1$
τ_{ver}	300	s	verification timeout (VERIFY state)

a) ApplyFilters(G, C): For every candidate GUTI $g \in G$, (i) map g to its current RNTI, (ii) measure $\max_i \Delta t_i$, and (iii) prune g if the gap exceeds τ_{gap}. The pruned set is returned to the FSM. If the current activity is *not* classified as continuous (e.g., web browsing), `ApplyFilters` bypasses the Δt check and returns the original set G unchanged.

E. Main Algorithm

Algorithm 1 presents the complete FSM-based GUTI identification procedure. The algorithm takes synchronized camera feed and RF signal as inputs, maintaining a candidate set G that is progressively refined through state transitions. The core loop continues until reaching either the DONE state with a successfully identified GUTI or the FAIL_SAFE state after exceeding the iteration limit.

The algorithm initializes with an empty candidate set and zero iteration counter. Each state executes specific operations: burst detection in SCAN, GUTI extraction in COLLECT, and candidate filtering in FILTER, verification in VERIFY. The iteration counter ensures termination, preventing infinite loops that could occur from persistent noise or interference. The corresponding filtering parameters and their default values, including τ_{gap}, Δt_i, and τ_{ver}, are summarized in Table VIII.

a) Helper Functions.: `AlignTimelines` prepares the sniffer and camera timelines for later matching by mapping both to a common wall-clock time domain rather than enforcing strict NTP-level synchronization. `DetectBursts` returns the set of RNTIs whose DL+UL volume within $[T - t_{th}, T + t_{th}]$ exceeds d_{th}. `ExtractGUTI` parses the `RRCConnectionSetup` message that allocates an RNTI to obtain the corresponding GUTI. `Verified` implements the predicate in (Appendix A-B5). `LogError` stores the current candidate set and iteration count for offline debugging.

The key functions called within the algorithm serve specific purposes: `DetectBursts` identifies RNTIs whose data activity exceeds the threshold; `ExtractGUTI` maps those RNTIs to their corresponding GUTI values through RRC messages; `ApplyFilters` eliminates candidates based on burst patterns and service characteristics; and `BurstMatch` determines whether the candidate GUTI is indeed the target by correlating additional data bursts with the established GUTI–RNTI mapping.

Complexity. Per iteration we scan at most n RNTIs and filter m candidates, yielding $O(n+m)$ work. With the iteration cap $k = 10$, the worst-case cost is $O(k(n + m))$.

Termination. The FSM monotonically advances the loop counter and aborts at $k = 10$; every state has a unique exit condition, hence the procedure always finishes in finite time.

F. FSM Complexity Analysis

Theorem 1 (Time Complexity)**.** *The FSM-based GUTI identification algorithm has worst-case time complexity of $O(k(n+m))$, where k denotes the maximum iterations, n represents the number of RNTIs, and m is the size of the GUTI candidate set.*

Proof. In each iteration, the algorithm processes at most n RNTIs in the SCAN state and filters at most m candidates in the FILTER state. Both operations are linear in the respective set sizes, yielding $O(n + m)$ work per iteration. With the iteration bound k, the total worst-case complexity is $O(k(n+m))$. □

G. FSM Robustness Guarantees

The FSM design provides the following robustness properties:

Lemma 1 (Termination)**.** *The algorithm terminates within finite time with either a valid GUTI or explicit failure.*

Proof. The iteration counter monotonically increases and is bounded by N_{iter}. Each state has defined exit conditions, preventing infinite loops. □

Key robustness features:

1) **Loop Prevention**: Bounded iterations
2) **Auto-Recovery**: $S_3 \rightarrow S_1$ on empty set
3) **Graceful Degradation**: FAIL_SAFE state
4) **Validation**: Verification

Appendix B
Formal Definition of No-Burst Intervals

Definition 4 (No-burst interval)**.** *Fix an RNTI and the burst detector $Burst(\textit{RNTI}, T)$. A closed interval $[a, b]$ is called a* no-burst interval *for this RNTI if*

$$\begin{aligned} b - a \ \geq\ 2t_{th} \quad &\textit{and} \quad Burst(\textit{RNTI}, T) = 0 \\ &\textit{for all } T \in [a + t_{th},\, b - t_{th}]. \end{aligned}$$

In other words, for every time T whose detection window $[T - t_{th},\, T + t_{th}]$ lies fully inside $[a, b]$, the accumulated DL/UL data remains below d_{th} and no data burst is detected.

Intuitively, a no-burst interval represents a time span in which no sliding detection window of width $2t_{th}$ contains sufficient DL/UL traffic to be classified as a burst. This definition excludes short quiet periods that cannot accommodate a full detection window.

List of Abbreviations

2G	2nd Generation
3GPP	3rd Generation Partnership Project
5G	5th Generation
BTS	Base Transceiver Station
CCTV	Closed-Circuit Television
CRI	Contention Resolution Identity
DCI	Downlink Control Information
DL	Downlink
EARFCN	E-UTRA Absolute Radio Frequency Channel Number
FN	False Negative
FoV	Field of View
FP	False Positive
FSM	Finite State Machine
GSMA	GSM Association
GUTI	Globally Unique Temporary Identifier
IMSI	International Mobile Subscriber Identity
LTE	Long-Term Evolution
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MME	Mobility Management Entity
MNO	Mobile Network Operator
M-TMSI	MME Temporary Mobile Subscriber Identity
NAS	Non-Access Stratum
NTP	Network Time Protocol
PCCH	Paging Control Channel
PDCCH	Physical Downlink Control Channel
PDU	Protocol Data Unit
PRB	Physical Resource Block
RF	Radio Frequency
RNTI	Radio Network Temporary Identifier
RRC	Radio Resource Control
SA	Stand-Alone (5G deployment mode)
SDR	Software-Defined Radio
SMS	Short Message Service
S-TMSI	SAE Temporary Mobile Subscriber Identity
UE	User Equipment
UL	Uplink
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
VoIP	Voice over IP
VoLTE	Voice over LTE