Towards Simultaneous Attacks on Multiple Cellular Networks

Alexander J. Ross
North Carolina State University
ajross6@ncsu.edu

Bradley Reaves
North Carolina State University
bgreaves@ncsu.edu

***Abstract*—Cellular network attack research has dramatically expanded its capabilities in the last decade, but threat models routinely assume an attacker who targets a single cell with a small number of moderately-priced software defined radios. In many settings, such as mass crowd surveillance, attackers seek to gain passive or active dominance over a given area that is virtually always served by multiple cells and network operators. To do so, the only method publicly available is to naïvely duplicate their hardware at extensive cost. This paper presents a preliminary analysis of the feasibility of using a single software defined radio to surveil multiple networks simultaneously. Our key insight is that an attacker is often interested in only a portion of transmissions in a cell, and by design cellular transmissions are rigidly and predictably scheduled. Our system, Intercellular, rapidly schedules a single radio to tune between cells, effectively multiplexing the downlink channels of cells together. We demonstrate that radio tuning time is quite low (around 100ms), radio clocks are sufficiently stable to skip synchronization when retuning, and that even when monitoring multiple cells a radio can quite accurately count the devices served by all cells under observation. In so doing, we open new research directions advancing the efficiency and broad applicability of cellular network attacks.**

***Index Terms*—Cellular Networks, Cellular Network Attacks**

I. Introduction

Cellular networks are critical infrastructures for nearly every aspect of life, and researchers have analyzed these networks to find vulnerabilities, demonstrate attacks, and propose fixes. The advent of software-defined radios (SDRs) and open-source air interface implementations democratized cellular attack research. SDRs in particular significantly lowered the cost barrier to entry, and they put cellular attacks within reach of academic researchers and attackers alike. Compared to the extreme costs of the custom hardware and software previously required for cellular research, SDRs are remarkably affordable.

SDRs are still costly, though, with the higher-end equipment costing thousands of dollars per unit. Accordingly, labs have few such radios for attack research, and attacks that target a single cell comprise practically all published work to-date. However, most areas are served by more than one carrier, each with multiple cells. This means that current state-of-the-art cellular research does not take into account the real-life cellular environment.

Unfortunately, the well-funded actors who actually conduct large-scale attacks apparently have no such limitations [1]. This leaves researchers who work on attacks and defenses "out-gunned" and unable to work within state-of-the-art threat models. Simply put, researchers live in a world with overlapping network coverage but only work one-cell-at-a-time.

In this work, we present a research vision of techniques that allow researchers to work in large multi-cell attack scenarios with laboratory-scale budgets. Our overarching approach is to "multiplex" a single radio across multiple cells to duplicate capability without duplicating hardware. Such an approach is feasible because most attackers seek only a fraction of overall traffic, such as temporary or permanent identifiers, paging and other signaling, or DNS requests. For this approach to work, the downtime of the radio during retuning and resynchronization must be small enough that the radio can capture the phenomena of interest.

To answer this critical question, we developed Intercellular, a proof-of-concept framework that identifies and counts the number of active devices in multiple cells simultaneously. In this preliminary analysis, we demonstrate the feasibility of achieving situational awareness of the number of nearby active devices across all cells. Simply knowing the number of active devices in a local area could be used for passive crowd estimation, no matter the carrier the subscriber is connected to. Identifying active devices within each cell is also a first step to performing a deeper analysis of the behavior of these devices in the network or identifying/tracking a device of interest.

Intercellular is a multi-downlink sniffer based on a heavily modified version of srsRAN [2] that is capable of synchronizing and monitoring multiple cells simultaneously. To facilitate maintaining synchronization with multiple cells simultaneously, we designed a rapid resynchronization algorithm that is used in place of traditional PSS/SSS-based synchronization. This algorithm uses the time of arrival of the last observed radio frame for that cell to calculate the time of arrival for the next radio frame. Such an approach is practical because LTE radio frames have a fixed periodicity and duration.

To identify active devices within each cell, we monitor the cell for uplink and downlink resource grants. Specifically, we monitor the downlink control channel for downlink control information (DCI). Intercellular monitors the DCI for unique radio network temporary identifiers (RNTIs) to identify all active UEs in each cell. The Intercellular DCI decoder is based on a heavily modified version of the FALCON DCI decoder [3].

We evaluated the probability that the Intercellular rapid resync algorithm successfully resynchronizes with the cell after

2023 IEEE Security and Privacy Workshops (SPW) | 979-8-3503-1236-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/SPW59333.2023.00040

DOI 10.1109/SPW59333.2023.00040

tuning back to it. Our experiments show that Intercellular could reliably achieve downlink synchronization, even when tuned off of the cell for up to 30 seconds. Additionally, our rapid resynchronization algorithm cuts the downlink resynchronization time in half, maximizing the coverage of the downlink decoder.

Since Intercellular cannot continuously monitor every cell simultaneously, only a fraction of the total subframes transmitted by the cell will be decoded. We quantified the coverage ratio for various cell configurations by checking if any active UEs were missed by the DCI decoder compared to a baseline trace. We observed that the coverage ratio decreased as the on-cell, and thus off-cell, sample time increased. We also observed that increasing the number of cells monitored simultaneously negatively impacted the coverage ratio.

This paper makes the following contributions:

- We present Intercellular, a technique for monitoring multiple 4G LTE cellular networks simultaneously.
- We demonstrate that Intercellular is capable of maintaining synchronization with multiple cells simultaneously.
- We evaluate the performance and accuracy of the Intercellular rapid resynchronization algorithm.
- We evaluate the coverage of our system using IQ samples from a generated cell for different configurations of sampling time and numbers of cells monitored simultaneously.

The remainder of this paper proceeds as follows. Section II will cover the technical background of LTE networks. Section III discusses the design of the Intercellular sniffer as well as our methodology. Section IV evaluates the performance of Intercellular. Section V provides a discussion of the limitations of Intercellular along with our plans for future work. Section VI overviews related works in cellular security. Section VII concludes.

II. Technical Background

In this section, we will discuss the major components and identities of the 4G LTE cellular network, paying particular attention to the physical layer. We will also cover essential identifiers used in 4G LTE cellular networks to address UEs.

A. LTE Cellular Network Overview

In LTE networks, two main components interact with the wireless channel, the eNodeB and the UE.

UE: The UE, also known as the User Equipment, is the mobile terminal subscribers use to access the cellular network. It communicates with radio towers, called eNodeBs, via a radio link known as the Evolved UMTS Terrestrial Radio Access Network (EUTRAN). Since UEs are power-constrained devices, they spend most of the time disconnected from the network, connecting only when a data transmission needs to take place.

eNodeB: The eNodeBs are the wireless base stations in a cellular network. Each eNodeB manages a single radio cell but may contain one or more sectors that may operate on different frequencies. The eNodeBs provide connectivity to the Evolved Packet Core (EPC). Resource allocations for both the downlink and uplink are explicitly scheduled by the eNodeB for all UEs on the network. The eNodeB is also responsible for sending paging messages to a UE to inform it of a call or data transmission.

B. Identifiers

LTE cellular networks use a mixture of permanent and temporary identifiers to address UEs on the network. Readers may be familiar with permanent identifiers such as the International Mobile Subscriber Identifier (IMSI) or temporary identifiers such as the Globally Unique Temporary Identifier (GUTI). However, in a properly configured network, temporary identifiers are broadcasted infrequently and permanent identifiers are never broadcasted outside of the initial attach.

In this work, we are primarily interested in the Radio Network Temporary Identifier, or RNTI, which is used to address a single UE within a single radio cell. There are multiple types of RNTIs, including the Cell RNTI (C-RNTI), Paging RNTI (P-RNTI), System Information RNTI (SI-RNTI), and the Random Access RNTI (RA-RNTI). The C-RNTI is an ephemeral identifier that identifies a particular UE for the duration of a single radio session. It is deallocated as soon as the UE disconnects from the network to switch off or handover to a different cell, or the network deallocates the UE if it was idle for more than about 10 to 30 seconds. The RA-RNTI is used when a UE first attempts to connect to the network and is quickly reallocated with a more permanent C-RNTI. The P-RNTI and SI-RNTI are reserved identifiers that distinguish paging and system information broadcasts destined for multiple UEs on the PDSCH.

C. Downlink Physical Layer of the Radio Access Network

The physical layer in LTE networks consists of the E-UTRAN, a radio link between the eNodeB and all UEs. The E-UTRAN Absolute Radio Frequency Channel Number (EARFCN) defines the center frequency of the cell. The system bandwidth can be as low as 1.4 MHz or as high as 20 MHz, and it can operate in either frequency division duplex (FDD) or time division duplex (TDD) modes.

The downlink consists of multiple physical channels and signals that are multiplexed together using Orthogonal Frequency Division Multiple Access (OFDMA). It is divided into frames that represent a 10ms period in time which are then further subdivided into subframes, each representing a 1ms period. The individual subcarriers are grouped into resource blocks, each consisting of 6 subcarriers and 6-14 symbols per subcarrier in a subframe depending on the Cyclic-Prefix (CP) length and the subcarrier spacing.

The multiplexed channels are assigned resource allocations within the frame depending on the type of channel, subframe number, the subcarrier and block number, and the system bandwidth. Some channels, such as the primary and secondary synchronization signals and the Physical Downlink Control CHannel (PDCCH), have fixed resource allocations in specific subframes within the radio frame. Dynamic resource allocations, such as user-plane data on the Physical Downlink Shared

CHannel (PDSCH), are explicitly signaled using the Downlink Control Information (DCI), which is scheduled on the PDCCH.

D. Downlink Synchronization

LTE uses two different synchronization signals, the PSS and the SSS, to facilitate UE downlink synchronization.

PSS: The Primary Synchronization Signal consists of one of three possible Zadoff-Chu sequences spanning the central six resource blocks. It is broadcasted every 5 ms in subframes 0 and 5 to provide a synchronization reference for a UE to blindly find the center frequency of the cell during the cell search procedure. The physical layer identity of the cell depends on the selected Zadoff-Chu sequence.

SSS: The Secondary Synchronization Signal consists of 168 M-Sequences spanning the central six resource blocks. It is used to identify whether the cell is using Frequency Division Duplexing (FDD) or Time Division Duplexing (TDD), detect the length of the cyclic prefix, and obtain radio frame synchronization. The position of the SSS encodes whether the cell is using FDD or TDD and determines the cyclic prefix duration when combined with the PSS. The SSS is broadcasted every 5 ms in subframes 0 and 5 for FDD or 1 and 6 for TDD. The SSS payload is changed for both transmissions to permit the UE to determine the radio frame boundary.

E. System Information

LTE also broadcasts system information in the MIB and two or more SIBs to configure UEs for network access or broadcast other control information to all UEs simultaneously.

MIB: The MIB is essential for configuring the radio interface on the UE. It conveys the system bandwidth, the Physical Hybrid ARQ Indicator Channel (PHICH), the current system frame number, and the number of transmit antennas used by the eNodeB.

SIB: LTE uses several System Information Blocks to convey system information about the cell. They are scheduled on the downlink with an SI-RNTI indication in the DCI. The payload within the SIB determines the type of the SIB and can convey information such as uplink configuration info, a list of neighboring cells, or even emergency alert broadcasts.

III. Intercellular Overview

The goal of Intercellular is to monitor multiple independent LTE cells simultaneously using only a single radio. We built the Intercellular LTE downlink sniffer on top of a heavily modified version of srsRAN [2] and FALCON [3]. In this section, we first describe the unique challenges of operating with more than one cell simultaneously. We then describe the process by which Intercellular acquires and maintains downlink synchronization. Finally, we describe the design of the DCI decoder.

A. Design Challenges

Monitoring multiple cells simultaneously presents a unique set of challenges that must be addressed. First, we need to maintain synchronization with multiple cells simultaneously without requiring continuous monitoring of each cell. srsRAN was not originally designed to maintain synchronization with multiple cells simultaneously. It does not support hot swapping cells, nor does it support synchronizing with the cell rapidly. While it is possible to use PSS/SSS-based blind synchronization every time we change cells, this will negatively impact the performance of our system since it increases the proportion of time that the decoder is idle.

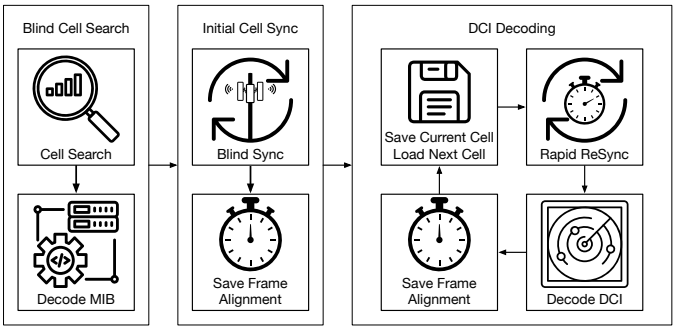


Figure 1. The Intercellular pipeline consists of 8 unique steps grouped into 3 distinct phases

Prior implementations of DCI decoders such as FALCON [3], IMDEA OWL [4], and C^{3}ACE [5] all operate on a single cell at a time. However, the Intercellular DCI decoder must decode the downlink control information (DCI) in each cell independently. The state of the DCI decoder must be carefully managed to prevent one cell from influencing the decoder in a different cell.

B. Downlink Synchronization

Intercellular has been designed to save the context for the current cell, including all active UEs, downlink time alignment, and system information required to quickly resynchronize with the downlink. The synchronization pipeline is shown in igure 1.

1) Achieving Initial Synchronization: When Intercellular is switched on, it tunes to each cell in the cell list and performs PSS/SSS-based cell search to detect whether a valid cell is present and decode system bandwidth. Intercellular then scans each found cell again to decode the Master Information Block (MIB) and establish frame alignment with the cell. Finally, a timestamp of the start of the last decoded radio frame is saved for use during rapid resynchronization.

2) Radio Tuning: An important metric to the design of Intercellular is the speed at which we can retune our radio to change cells as this will impact the optimal speed at which we sample cells.

In Software Defined Radio systems, two types of latency affect the amount of time required to retune the radio. This includes the latency of the host system processing and transmitting tuning requests to the SDR and the retuning time of the analog front end within the SDR. We can mask out the tuning time for the host if we know the exact moment we want to change frequencies in advance. However, the analog

front-end tuning delay must be taken into account as it impacts the maximum coverage achievable by the system.

For USRP SDRs, the USRP driver provides an interface to schedule a command to run at an exact time by providing a `time_spec` with the command. Alternatively, `set_command_time` can be used to specify the time at which to execute the next command sent to the radio.

3) Rapid Resynchronization: To minimize the amount of time spent in tuning and synchronization, we designed Intercellular to save synchronization information which permits rapid resynchronization with the downlink. Given that LTE radio frames are exactly 10ms in length, we theorized that we could calculate the start time of a new radio frame if the start time of a previous radio frame is known.

At the end of each monitoring period, Intercellular saves a timestamp corresponding with the start of the last decoded radio frame. When Intercellular tunes back to the cell, it first instructs the radio to tune to the new center frequency and waits for the tune to complete. Intercellular then calculates the time at which the next radio frame should occur based on the current radio timestamp. A timed receiver stream start command is then issued to the radio so that the first sample returned corresponds with the start of the next radio frame. We modified the radio core in srsRAN to support issuing a timed receive stream start command to the radio.

Frame and subframe synchronization is checked in every subframe during downlink decoding. If frame synchronization is lost or the rapid resync procedure fails, Intercellular will automatically fall back to performing blind PSS/SSS-based synchronization. At the end of the monitoring period, Intercellular will save a new frame alignment time stamp and retry rapid resync in the next monitoring period.

C. Identifying Active Devices

1) Counting Active Devices using Downlink Control Information: To identify active UEs within each cell, we designed Intercellular to monitor the Downlink Control Information (DCI) in the PDCCH for resource allocations.

In LTE networks, the DCI is an essential component for coordinating data transmissions in the network. The DCI conveys resource grants that inform UEs of transmission of downlink transmissions destined for that device. It also informs the UE of uplink transmission opportunities. All communications are scheduled by the eNodeB, including the uplink. The UE cannot arbitrarily choose when to transmit data to the network.

We implemented the DCI decoder using a heavily modified version of the FALCON DCI decoder [3]. The FALCON DCI decoder and physical layer subframe worker were ported to support running with hot swap cell contexts.

Whenever the UE communicates with an eNodeB, it is assigned a C-RNTI by the eNodeB, which lasts for the duration of the radio session. A DCI resource grant, however, does not explicitly signal the intended recipient. Instead, the CRC for each DCI control message is scrambled by the C-RNTI to identify the intended recipient. DCI messages are rate matched, stretching the number of bits into the number of bits in one or more control channel elements (CCEs). This makes it impossible to determine which DCI format was used quickly.

When Intercellular detects a DCI transmission within a subframe, it decodes the indication in every possible DCI format and aggregation level. It then descrambles the CRC of each DCI format candidate to recover candidate C-RNTIs. The DCI format candidates are then filtered out based on rules such as whether a particular RNTI is permitted for a particular format. Finally, candidate RNTIs are added to a histogram. If a particular RNTI exceeds the histogram count for a specific period of time, then the RNTI is marked as active.

By monitoring the DCI for C-RNTIs, we can obtain an instantaneous count for all active UE's in a local area; that is, they are in the `RRC-CONNECTED` state and have active data transmissions. Intercellular cannot identify UEs that are switched off, idle or otherwise inactive for the duration of the scan. Therefore, this will give us a lower bound on the actual number of UE's currently within the cell.

When a UE has been inactive for a significant period of time, the eNodeB will deallocate the UE and transition it to the `RRC-IDLE` state. The time before an inactive UE transitions from `RRC-CONNECTED` to `RRC-IDLE` is defined in the dataInactivityTimer at the eNodeB. The exact value is carrier-dependent but typical values range from around 10-30 seconds. Intercellular marks an active RNTI as inactive if it has not seen a DCI allocation for that RNTI in over 30 seconds.

Counting UEs based on observed DCI messages can only approximate the true number of active UEs in the cell. The true number of active devices could be overestimated if a UE leaves the cell (e.g., handover or switch-off) but the inactivity timer has not expired yet. Simultaneously, the true number of active devices could also be underestimated if a new device joins the cell, but Intercellular has not observed enough DCI allocations to mark the RNTI as active. Additionally, since Intercellular does not monitor a cell continuously, a UE could enter and leave the cell in between monitoring periods, thus causing that RNTI to never be marked as active. Therefore, the number of active devices reported by Intercellular is only an approximation of the true number of active devices within the cell.

IV. Experimental Evaluation

In this section, we evaluate the performance of Intercellular. We start by describing the hardware we implemented Intercellular on as well as a research LTE network powered by srsRAN. We then describe the experiment we performed to characterize how quickly a USRP B210 radio can be returned. Next, we perform an experiment measuring the stability and accuracy of our time-based fast synchronization algorithm. Finally, we evaluate the performance of our system when monitoring multiple cells simultaneously.

A. Research LTE Network

Intercellular was executed on a Dell Precision 3650 workstation equipped with an Intel Core i7-11700 and 32 GB of

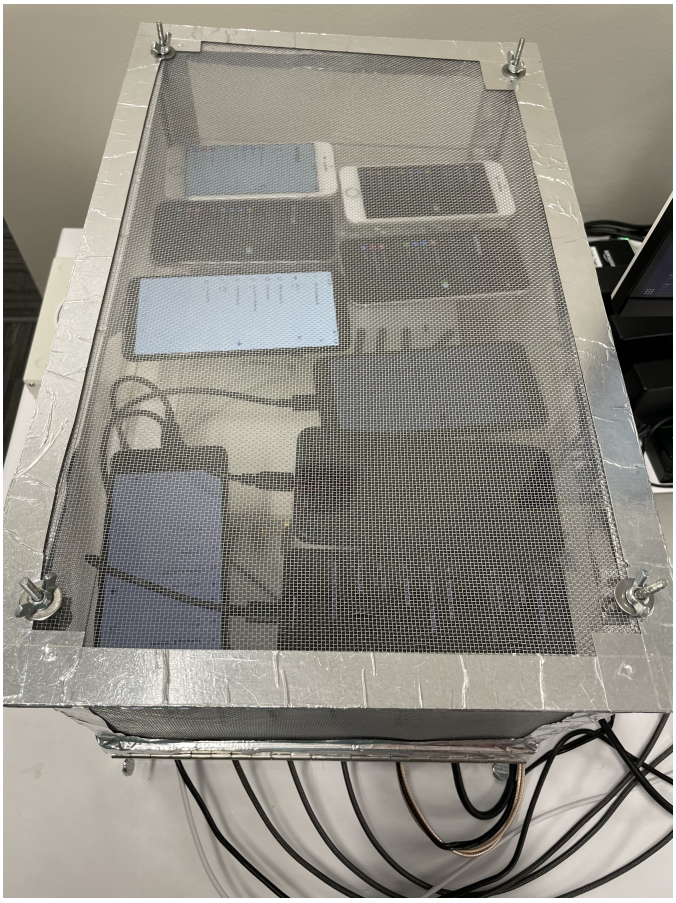


Figure 2. Faraday cage for our active LTE experiments.

RAM. A USRP B210 SDR served as the radio to sniff the LTE downlink for both our research and several commercial LTE networks. In our experiments, we used an Ettus Research Octoclock-G CDA-2990 clock distribution amplifier with an integrated GPS disciplined oscillator (OCXO) to provide a steady time and frequency source. An internal GPSDO for the USRP B210 can also be used to provide the time and frequency source. While the use of a GPSDO is not required for Intercellular to operate, the use of a GPSDO, whether internal or external, is recommended as it will improve the accuracy of the Intercellular rapid resynchronization procedure, resulting in fewer fallbacks to blind PSS/SSS-based synchronization. We chose to use the external GPSDO as we did not have individual GPSDO modules for both of the radios on hand.

Several of our research experiments require the transmission and reception of known LTE network signals. We used srsRAN, an open-source LTE network stack, to serve as our research network. We ran this network on a Dell OptiPlex 9020 equipped with an Intel Core i7-4770 and 32 GB of RAM. A second USRP B210 SDR was used to broadcast the LTE network. This USRP was also connected to the GPSDO to improve the frequency accuracy of the broadcasted cell. In this preliminary exploration, we used an external clock to ensure that oscillator instability does not affect our experiments. We hypothesize that this step was overly cautious, and will evaluate whether an external clock is needed in future work.

Since some of our research experiments required the transmission of LTE network signals over the air, we took precautions to ensure that we did not cause any interference with commercial mobile network operators or any other wireless service. All experiments were performed within a Faraday cage, and we broadcasted at the minimum power required. We configured srsRAN to broadcast on 948 MHz with an uplink of 903 MHz (EARFCN 3680), which is in LTE band 8. This band was chosen because it is partially within the 900 MHz US Industrial, Scientific, and Medical band. Wireless devices sharing this band should be resilient to any interference that we may inadvertently produce. Additionally, we tested our homebrew Faraday cage setup and found no significant RF emissions outside the cage. The Faraday cage setup is depicted in Figure 2.

We used up to nine unique phones in our research network, including four Android phones and five iPhones. Custom SIM cards from Osmocom were programmed to facilitate connecting this UE to the srsRAN network. We used the `scrcpy` utility to remotely control all Android phones inside the Faraday cage.

B. Benchmark Radio Tuning Performance

In this experiment, we are interested in empirically determining the time it takes for the radio to change frequencies and stabilize. This tuning delay directly influences the maximum coverage that can be achieved as no frames can be decoded during each tune.

USRP radios have a two-stage tuning process. The first stage tunes the analog front end to a frequency close to the desired frequency to convert RF to a predetermined intermediate frequency (IF). Once the analog front end stabilizes, a "digital tune" in the hardware DSP chain converts the IF to baseband samples. USRP radios can accept a reference 10 MHz clock to stabilize the internal timebase and serve as a reference input to the phased lock loop (PLL) used in the analog front end. We performed our tuning experiments with both the internal clock and the external reference clock to examine whether there was any significant difference depending on the clock source.

We chose a starting frequency of 700 MHz and an ending frequency of 900 MHz, giving us a total tuning delta of 200 MHz. These frequencies were selected to ensure the radio had to perform an analog front-end retune, as the tuning delta is larger than the analog front-end bandwidth. The radio is configured to use a sampling rate of 2 MSps and a bandwidth of 200 KHz.

Finding 1: We observed an average tuning delay of 106.407ms with a standard deviation of 1.470ms. When an external clock was used, the average tuning delay was 105.506ms with a standard deviation of 0.311ms.

C. Fast Resynchronization

Maintaining synchronization with every LTE cell being monitored is critical to decoding the LTE resource grid successfully, thus permitting the recovery of DCI indications from the PDCCH. To maximize the coverage of every cell

being monitored, we need to minimize the amount of time that is spent not listening to the cell. Traditional PSS/SSS-based synchronization uses synchronization signals broadcasted by the eNodeB to determine time and frame alignment. This procedure, however, takes a significant amount of time to complete, thus reducing our maximum achievable coverage of each cell.

To solve this issue, we implemented a rapid resynchronization algorithm that exploits the fact that LTE radio frames are exactly 10ms in length. This permits the starting time and sample of the next radio frame to be calculated in advance and issue a timed receive stream start command to the radio such that the first sample returned corresponds with the first sample of a radio frame. Provided that a stable and accurate timebase is used, it should be possible to accurately calculate starting time and sample of the next radio frame for any duration of time spent off of the cell.

To evaluate the performance of the Intercellular rapid resync algorithm, we performed two experiments. The first experiment evaluates the amount of time saved using Intercellular rapid resync compared to srsRAN's built-in traditional PSS/SSS-based synchronization. Our next experiment evaluates the accuracy of fast resynchronization for different durations of time spent off of the cell. In both of our experiments, we used srsRAN to broadcast the cell in band 8.

1) Intercellular Rapid Resynchronization Performance compared to traditional PSS/SSS based syncrhonization: To evaluate the performance of the Intercellular fast resynchronization algorithm, we implemented an experiment mode that repeatedly performs both the Intercellular rapid resync procedure and srsRAN's synchronization procedure. The amount of time required to tune the radio to the cell, synchronize with the cell, and check the position of the PSS reference signal is recorded in each run. We broadcasted our reference cell using srsRAN and a second USRP B210 radio in our faraday cage. Both radios utilize a stable time and frequency source provided by an external GPS-disciplined oscillator.

Intercellular first performs blind cell search and synchronization with our testbed cell and decodes the MIB to set the cell context. After decoding the MIB, Intercellular resynchronizes with the cell using blind PSS/SSS synchronization at the full bandwidth of the cell. A timestamp corresponding with the start of the radio frame is then saved for use with the rapid resynchronization procedure.

Intercellular then tunes off of the cell to another frequency to simulate a tune to a different cell. Since our research cell broadcasts on 948 MHz, we chose to use a frequency of 2.4 GHz to simulate tuning between bands. We then request samples from the radio to simulate receiving from another cell, but we do not decode them. After a delay of 500ms, Intercellular initiates a tune request to tune the SDR back to the original frequency. A timestamp is also taken right before the tune request is sent to the radio.

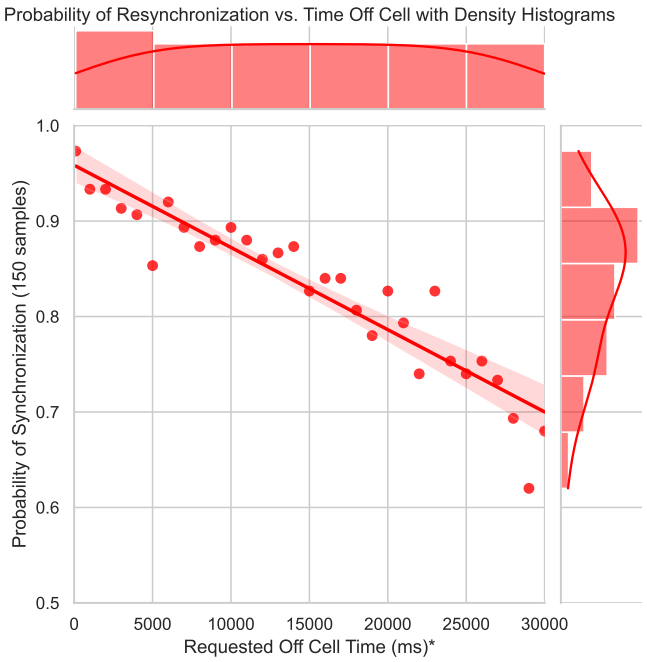
If Intercellular rapid resync was selected, the procedure is executed to achieve frame alignment with the cell. Otherwise, srsRAN's PSS/SSS-based sync procedure is executed instead. After synchronization completes, a subframe worth of symbols is requested from the radio. If synchronization was achieved, this will be the first subframe of the radio frame. This subframe is then checked for the presence of the PSS. If the PSS is present, a timestamp is taken to mark the end of the synchronization procedure. If the PSS is not present, blind PSS/SSS-based synchronization is repeated until the PSS is detected successfully.

Finding 2: Intercellular's rapid resync procedure is 191% faster than srsRAN's built-in blind PSS/SSS-based synchronization procedure. We observed that a complete radio tune using Intercellular rapid resync had an average duration of 176.45ms over 100 samples. Out of this duration, 171.63ms was spent waiting for the radio to tune while an additional 4.1ms was spent running the tuning algorithm and checking the position of the PSS. The srsRAN sync procedure took significantly longer with an average delay of 336.18ms over 100 samples. 171.54ms was spent waiting for the radio to tune while an additional 163.89ms was spent searching for the PSS. Intercellular rapid resync saves 159.73ms per tune over srsRAN's resync procedure, cutting the amount of time required to obtain the first usable subframe by nearly half.

2) Intercellular Rapid Resynchronization Accuracy: To evaluate the accuracy of the Intercellular rapid resync algorithm, we implemented an experiment mode in Intercellular that repeatedly performs rapid synchronization with our research cell and checks if synchronization succeeded. We utilized a similar procedure to the Intercellular rapid resync performance experiment. Intercellular first performs a blind cell search to find the cell and decode the MIB to set the cell context. After decoding the MIB, it then resynchronizes with the cell at the full system bandwidth and saves a time stamp corresponding with the start of the last decoded radio frame. Intercellular then instructs the SDR to tune away from the cell to another band and wait for the duration of the requested off-cell.

After the off-cell time had elapsed, Intercellular issues a tune request to tune the radio back to the cell and waits for the tune to complete. Intercellular then calculates the time at which the next radio frame should occur based on the timestamp indicating when the previous radio frame occurred plus a small offset to ensure that the radio can process the command in time. A timed receive stream start command is then issued to the SDR and the location of the PSS is checked. If the PSS decodes successfully, resynchronization is considered to have succeeded. Before repeating the experiment for another sample or duration off-cell, blind synchronization is performed to ensure that a fresh timestamp is used the next time Intercellular tunes back to the cell.

Finding 3: We observed that Intercellular could reliably calculate the start time of the next radio frame and achieve downlink synchronization. Intercellular had better performance for shorter off-cell durations, with the odds of achieving synchronization above 90% if off-cell for less than 5 seconds. Even when Intercellular was off-cell for up to 30 seconds, we were still able to reliably resynchronize with the cell 60% of the time. A graph of the resynchronization probability is shown



*Requested off cell time is the duration between when the radio is tuned off of the cell to when a tune request is issued to tune back to the cell. Actual time spent off cell is greater due to analog front-end tuning delay

Figure 3. Probability of successfully synchronizing decreases as the amount of time between monitoring periods increases.

in Figure 3.

D. Intercellular DCI Decoder Coverage

The ultimate goal of our system is to identify all active UEs across all cells in a local area. To count the number of UEs active in the local area, we decode the Downlink Control Information (DCI) and descramble the CRC to recover the Radio Network Temporary Identifier. Given that our system monitors each cell for only a short period of time, the DCI decoder will not have access to every DCI indication present in the cell. This will cause some active UEs within the cell to go undetected. The ratio in the number of UEs detected in a multi-cell system vs a single-cell system defines the achievable coverage for that system configuration.

To evaluate coverage, we needed to capture a ground truth trace from a cell with a significant number of UEs. In an ideal scenario, we would perform an experimental analysis using traces captured from commercial cellular networks. However, capturing a trace from a live commercial cell was not an option because it is impossible to obtain ground truth data from the carrier.

Another option would have been to broadcast our own network using srsRAN. We had a total of 9 phones that were compatible with our network. However, this sample size is too small, nor would it represent the load experienced by a typical cell. While we could have repeatedly connected and disconnected these 9 UEs to observe if any were missed by Intercellular, we also observed that srsRAN is unstable when data is being transmitted to and from the UE and would randomly hang briefly during data transmissions. This would cause subframe samples to arrive late to the radio which in turn caused every UE connected to experience radio link failure (RLF). Although Intercellular is tolerant to synchronization errors when capturing directly from the radio, it is not tolerant to synchronization errors when reading from an IQ file due to the way srsRAN's `ue_sync` library is implemented.

To solve both of these issues, we elected to generate ground truth data by simulating a cell in MATLAB instead of performing a live experiment. To generate an ideal trace, we utilized the `lteRMCDLTool` to generate six-hundred thousand subframes, equating to a ten-minute sample duration. The generator was configured to use 6 PRBs and up to 20 simulated UEs could be active at once. 235 randomly allocated UEs, each with a random amount of downlink data, were simulated in this cell. Each simulated UE contained a Radio Network Temporary Identifier and a counter to count the number of subframes that the UE was allocated for downlink reception.

UEs were sequentially allocated at random points in the trace using a weighted random variable. The more simulated UEs that were active at once, the less likely a new UE would be allocated up to a limit of 20. Each simulated UE could receive data in 500 to 10,000 subframes before being deallocated.

In each subframe, each simulated UE had a chance to be deallocated if they had been allocated to receive data in at least 500 subframes. The chance for the UE to be deallocated increased proportionally as the number of allocated subframes increases. A UE would automatically be deallocated if it received 10,000 subframes worth of data.

The UE scheduling allocations were simulated by picking UEs to schedule in a subframe at random. If a UE was scheduled for a subframe, that subframe contained a DCI Format 1 allocation with the RNTI of that UE. If a particular UE was scheduled to receive data in the previous subframe, the same UE had a 99.7% chance to be allocated in the next subframe. If the UE was not selected to receive data again, a new UE had a 10% chance to be allocated in the next subframe. If no UE was allocated for a particular subframe, that subframe carried an SI-RNTI DCI allocation instead.

To perform this experiment, an experiment mode was added to Intercellular to simulate tuning on and off cell when reading from a file. We define on-cell sampling time as the amount of time spent actively listening to a cell and off-cell time as the amount of time spent tuned away from the cell. The Intercellular DCI decoder was then executed on the same trace to generate a DCI trace for different configurations of cell sampling time and the number of monitored cells. Finally, we compared the DCI trace with a baseline single-cell trace to count the number of unique RNTIs detected vs the number expected.

To quantify the coverage ratio, we first ran Intercellular in single-cell mode on the trace generated by MATLAB to generate a baseline DCI trace. We then re-ran Intercellular configured to simulate tuning between multiple cells on the same reference trace. We configured Intercellular to simulate monitoring between two and fifty cells simultaneously and for

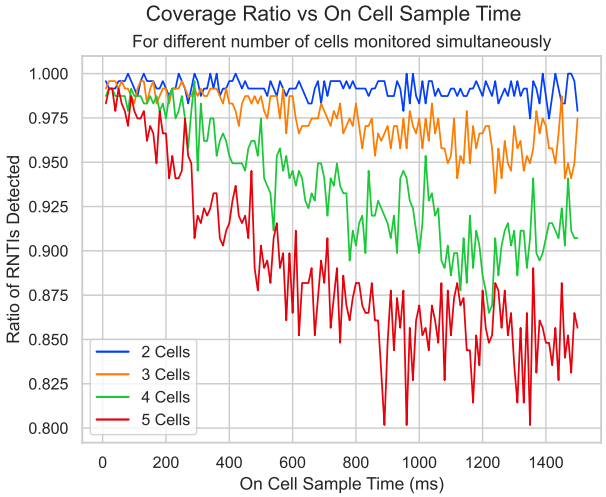


Figure 4. Coverage decreases as the amount of time spent listening to each cell increases.

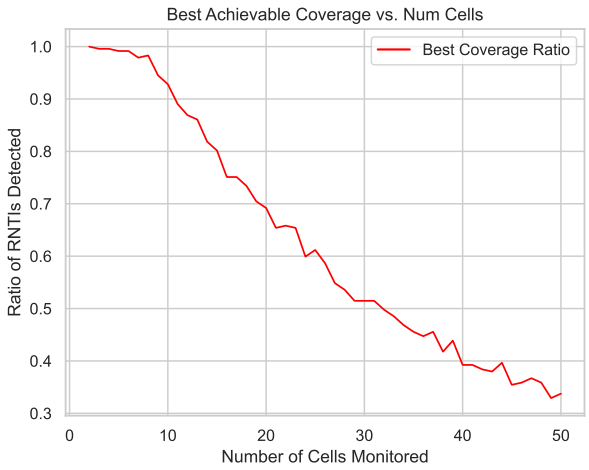


Figure 5. Coverage decreases as the number of cells monitored simultaneously increases.

on-cell sample times ranging from 10ms to 1500ms in 10ms steps.

The DCI trace for a particular number of cells and on-cell sample time was then compared to the baseline DCI trace to determine the ratio in the number of UEs detected. We simulated monitoring multiple cells by decoding the DCI for the duration of the monitoring period and then skipping several radio frames before resuming decoding. The number of radio frames that were dropped was determined by the monitoring period and the number of cells in the system. We simulated the radio tune by dropping an additional 12 radio frames, corresponding to 120ms of time, for each time the cell switched.

DCI allocations are treated slightly differently in this experiment as opposed to the live version of the Intercellular decoder. RNTI allocations do not expire in this experiment. Coverage is quantified by computing the ratio of the number of unique RNTIs observed by Intercellular to the ground truth data set.

Finding 4: We observed decoding fewer radio frames per sampling occasion improved the odds that an active device in the cell would be detected, as shown in Figure 4. Interestingly, monitoring only two cells resulted in nearly the same coverage as listening to a single cell for the sample times we tested. Coverage performance began to decline significantly as time increased once the number of cells monitored simultaneously increased beyond 3 cells.

Finding 5: We also observed that increasing the number of cells monitored simultaneously negatively impacted the coverage of the cell, as shown in Figure 5. In the above graph, we took the best achievable coverage for every cell configuration. Once the number of cells monitored simultaneously increased beyond 8, the performance began to decline sharply. At 20 cells monitored simultaneously, only about 70% of the active UEs were detected. This decreased to 50% of active UEs being successfully detected when monitoring 32 cells at the same time.

Our findings show that it is practical to monitor a small number of cells simultaneously using a single radio with a minimal drop in coverage.

V. Discussion

In this section, we provide a discussion of the limitations of the Intercellular system and provide a preview of future work.

A. Intercellular Limitations

There are several known limitations to the Intercellular system.

Intercellular can only monitor a single radio cell at a time, even though the radio is rapidly tuned between stations. No cell is continuously monitored, which can result in UEs being missed if they were not active during any of the monitoring periods for that cell. The active UE counts outputted by Intercellular are only estimates and may or may not represent the actual number of active UEs in the cell.

Intercellular cannot instantly switch cells as the radio takes a significant amount of time to retune. This leads to significant gaps in data collection, especially if a short monitoring period is used. This issue can be mitigated by using a second physical radio (or RX channel) and switching between them so that one radio (or channel) is always listening while the other is tuning. An alternate option is to receive the entire band at once and split each cell using DSP techniques, but this only works for bands that have a bandwidth smaller than the maximum analog bandwidth of the radio.

Additionally, DCI messages of different formats are rate matched into one, two, four, or eight fixed-size control channel elements. This rate matching destroys the ability to recover the DCI format quickly. While a CRC is attached to the DCI, it is scrambled with the RNTI of the destination device. Real LTE UEs can detect the correct format by checking if the CRC verifies for a particular format. Intercellular has to blindly guess which format is used as there always exists an RNTI which

makes the CRC check pass, even if the wrong format is used. A histogram is used to guess which RNTIs are likely active, but this approach still is susceptible to false positives.

B. Future Work

The overall goal for this line of research is to develop a system that can efficiently interact with multiple cells simultaneously with the minimum number of radios required. We currently envision enhancing Intercellular to support better UE fingerprinting and tracking, operation across multiple cellular generations, uplink synchronization and monitoring, and performing cellular attacks across multiple cells simultaneously.

1) Improving Radio Performance: There are several optimizations we intend to explore that will enhance the performance of Intercellular. Some SDRs have a second RX input. If this input is completely independent of the first, then we may explore switching between RX inputs so that the radio is always receiving from at least one channel. Additionally, for some bands, the total bandwidth available is less than the maximum analog bandwidth of a USRP B210 (56 MHz). We may explore using DSP techniques to receive multiple closely spaced cells simultaneously. Closely spaced cells could then be paired to minimize the number of analog tunes required. SDRs with larger analog bandwidths can also be used to increase the number of cells that can be decoded simultaneously. Finally, it is possible to manually cache `tune_result_t` parameters from the radio for later use in constructing a manual `tune_request_t`. We may explore whether caching these tune requests will optimize the analog tuning performance of the radio.

2) Enhanced Monitoring, Fingerprinting, and Tracking: Our initial prototype system is the starting point for device fingerprinting and tracking. Currently, Intercellular can decode the PDCCH from each cell to capture the downlink control information (DCI). We intend to expand the system to use the data collected from the PDCCH to facilitate capturing of PDSCH traffic for specific UEs to perform device fingerprinting and/or tracking. Existing research on passive UE fingerprinting and tracking exclusively focused on a single cell at a time to provide either cell-wide traffic analysis or cell area tracking. We plan to apply Intercellular to existing research to implement a system that can fingerprint UE activity and/or track all cellular devices in a local area, regardless of the carrier that the UE is using.

3) Operating Across Generations: Most areas are served by multiple network generations. Even though some carriers are starting to sunset their 2G and 3G networks, 4G LTE and 5G networks will coexist for years to come. Additionally, 4G LTE and 5G networks are designed to interoperate between each generation. A UE could handover between network generations for a variety of reasons such as signal quality, data throughput, and power requirements. This means that obtaining situational awareness of every device in a local area requires interoperating with multiple network generations. We plan to initially start with adding 5G low band (up to 6 GHz) support to our system. mmWave support will be added in the future when SDRs capable of operating in cellular mmWave become commonly available. Support for earlier generations may be added to facilitate the development of cross-generation attacks.

4) Uplink Synchronization: Our initial prototype synchronizes only with the downlink of each cell, collecting DCI indications passively over the air. While this simplifies the design of the system, it limits the system to only performing passive traffic analysis. Synchronizing with the uplink permits Intercellular to perform passive uplink monitoring with enhanced UE fingerprinting and tracking capabilities. Implementing uplink synchronization also opens the door to performing active attacks against multiple cells simultaneously.

5) Attacking Multiple Cells Simultaneously: Attacking multiple cells simultaneously is the ultimate goal of this line of work. Existing attack research has exclusively focused on attacking a single cell at a time. While this is useful if the goal is to disrupt operations for a single cell, one might want to attack an area instead, regardless of the carrier the subscriber is connected to. We intend to revisit previous cellular attack research to identify ones that can be applied to multiple networks simultaneously. For each of these attacks, we will demonstrate that they function across multiple cells as well as measure the effectiveness of the attack compared to a single-cell scenario.

VI. Related Work

Prior research on cellular networks has revealed a variety of attacks that impact the security, privacy, and/or availability of cellular networks. Most attacks focus on leaking information about a user or the network, while others seek to disrupt legitimate communications or send unwanted calls and text messages to users.

A. Physical Layer

The physical layer in any generation of a cellular network is the Radio Access Network (RAN), a wireless link between UEs and base stations. Prior work on the physical layer ranged from jamming or tampering with control plane traffic to performing a denial of service attack [6], [7], [8], to building analysis tools to detect anomalies [9], [10], or track cell load [11]. Another interesting use for the physical layer is for estimating the crowd density through classifying changes in the referenced signal received power on LTE [12] or the received signal strength in WiFi [13].

B. Layer 2 Fingerprinting

Prior work in a related field demonstrated that website traffic is fingerprintable even if the traffic is encrypted [14], [15], [16], [17], [18]. When these same methods were applied to cellular layer two traffic, researchers were able to successfully fingerprint the Alexa Top 50 websites despite the traffic being encrypted over the air at the PDCP sublayer [19], [20]. Encrypted website fingerprinting can also be used to classify and monitor the behavior of IoT devices [21].

Layer 2 fingerprinting attacks can also leak the identity of a subscriber, which can later be used to reveal the subscriber's location. The network can be stress tested to produce paging

requests that can be used to extract the GUTI [19], [20] or even the IMSI [22]. Kotuliak et al. take this further by demonstrating that user locations can be inferred by monitoring the UEs uplink timing alignment [23].

C. Paging Channel

When a UE is not actively communicating with the network, it periodically monitors the paging channel to check for paging requests to save power [22]. Paging attacks involving the GUTI/TMSI exploits have also led to numerous location leaking attacks, which can identify whether a user is within a radio cell [24], [25], [22], [26], [27], [28], [29]. The paging attack can be extended to session linking by monitoring uplink received power to identify if a stationary UE is in approximately the same location [30]. Hong et al. analyzed when and how the GUTI is reallocated in a sample of several cellular networks and identified that the identifier is often not reallocated frequently enough and/or the identifier is reallocated such that the next GUTI can be derived from the previous GUTI [24]. The GUTI/TMSI can be revealed by stress testing the network by either placing phone calls or sending messages to the UE and listening to the paging channel for all possible paging messages [24], [31]. Other attacks on the paging channel consist of full denial of service attacks [32], [33], [34], transmitting fake paging messages [32], [33], or impersonating the end UE to receive phone calls, SMS, or data services as another subscriber [32], [35].

D. Eavesdropping / Man in the Middle

Extensive research has been conducted on man-in-the-middle devices that permit an adversary to intercept messages, alter messages, deny services, or even reveal the subscriber's identity. IMSI catchers are cell site simulators that attempt to trick UEs into connecting to a fake 2G network to capture the IMSI or eavesdrop/tamper with voice, sms, or data traffic [36], [37], [38], [36], [39], [40]. Weaknesses in the authentication of UEs and Base Station Transceivers (BTS) in 2G permits unauthorized connections to be established, even in newer network generations [41], [42]. IMSI catchers can be fingerprinted by detecting unusual protocol requests, such as downgrades to insecure ciphers [37], [36]. Echeverria et al. proposed PHOENIX to help detect attacks or unsafe network configurations by monitoring device-side LTE traffic and performing signature-based behavior classification [43]. 5G attempts to address the privacy concerns of IMSI catchers by transmitting a Subscriber Concealed Identifier (SUCI) instead of the Subscriber Permanent Identifier (SUPI) [44]. Other notable attacks include eavesdropping VoLTE calls [45], impersonation and session hijacking [46], [32], and tampering with data transmitted over the air [47], [19].

E. Cellular Network Protocol Analysis

The cellular network protocol for LTE and 5G has recently been under increased scrutiny for vulnerabilities. Several tools were developed to examine the LTE and 5G network protocols to identify unsafe procedures that could expose a vulnerability that attackers could exploit [48], [35], [49]. Protocol vulnerabilities were identified in the paging procedure, the attach and detach procedure, and the authentication and key agreement [50], [35], [49] and when transitioning between network generations [41]. In 5G, network slices can be hijacked by a rogue slice owner to deny service on the network [51].

VII. Conclusion

With the creation of Intercellular, we demonstrated a technique for monitoring multiple independent LTE cells simultaneously using a single radio. Although our system cannot truly monitor all cells simultaneously, it exploits the tight and predictable scheduling in LTE networks to maintain synchronization and capture information of interest in each cell. A significant portion of time is spent waiting for the radio to tune and synchronize rather than monitoring a cell. Despite these shortcomings, intelligent use of the radio can yield similar performance to continuous cell monitoring, despite listening only a fraction of the time. Expanding Intercellular to decode additional traffic channels besides the PDCCH extends the capabilities of the decoder to perform additional traffic analysis and even attacks against all cells in a local area.

References

[1] "Government Cellphone Surveillance Catalogue," Dec. 2015. [Online]. Available: https://theintercept.com/document/2015/12/17/government-cellphone-surveillance-catalogue/

[2] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srsLTE: an open-source platform for LTE evolution and experimentation," in *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*. New York City New York: ACM, Oct. 2016, pp. 25–32.

[3] 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)*. Waikoloa, HI, USA: IEEE, Dec. 2019, pp. 1–7.

[4] 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 - ATC '16*. New York City, New York: ACM Press, 2016, pp. 25–30.

[5] R. Falkenberg, C. Ide, and C. Wietfeld, "Client-Based Control Channel Analysis for Connectivity Estimation in LTE Networks," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. Montreal, QC, Canada: IEEE, Sep. 2016, pp. 1–6.

[6] F. Girke, F. Kurtz, N. Dorsch, and C. Wietfeld, "Towards Resilient 5G: Lessons Learned from Experimental Evaluations of LTE Uplink Jamming," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. Shanghai, China: IEEE, May 2019, pp. 1–6.

[7] M. Lichtman, R. P. Jover, M. Labib, R. Rao, V. Marojevic, and J. H. Reed, "LTE/LTE-A jamming, spoofing, and sniffing: threat assessment and mitigation," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 54–61, Apr. 2016.

[8] 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*, ser. MobiCom '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 743–755, event-place: Sydney, NSW, Australia.

[9] I. Samy, X. Han, L. Lazos, M. Li, Y. Xiao, and M. Krunz, "Misbehavior Detection in Wi-Fi/LTE Coexistence over Unlicensed Bands," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022.

[10] S. Kumar, E. Hamed, D. Katabi, and L. Erran Li, "LTE radio analytics made easy and accessible," in *Proceedings of the 2014 ACM conference on SIGCOMM - SIGCOMM '14*. Chicago, Illinois, USA: ACM Press, 2014, pp. 211–222.

[11] D. D. Chirkov, A. K. Gaysin, and I. P. Ashaev, "LTE Cell Load Estimation Based on DCI Message Decoding," in *2021 Systems of Signals Generating and Processing in the Field of on Board Communications*. Moscow, Russia: IEEE, Mar. 2021, pp. 01–06.

[12] S. Di Domenico, M. De Sanctis, E. Cianca, P. Colucci, and G. Bianchi, "LTE-based passive device-free crowd density estimation," in *2017 IEEE International Conference on Communications (ICC)*. Paris, France: IEEE, May 2017, pp. 1–6.

[13] S. Georgievska, P. Rutten, J. Amoraal, E. Ranguelova, R. Bakhshi, B. L. de Vries, M. Lees, and S. Klous, "Detecting high indoor crowd density with Wi-Fi localization: A statistical mechanics approach," p. 26.

[14] J. Hayes and G. Danezis, "k-fingerprinting: A Robust Scalable Website Fingerprinting Technique," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 1187–1203.

[15] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1928–1943, event-place: Toronto, Canada.

[16] K. Kohls and C. Pöpper, "DigesTor: Comparing Passive Traffic Analysis Attacks on Tor," in *Computer Security*, J. Lopez, J. Zhou, and M. Soriano, Eds. Cham: Springer International Publishing, 2018, vol. 11098, pp. 512–530, series Title: Lecture Notes in Computer Science.

[17] M. Nasr, A. Bahramali, and A. Houmansadr, "DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. Toronto Canada: ACM, Jan. 2018, pp. 1962–1976.

[18] R. Overdorf, M. Juarez, G. Acar, R. Greenstadt, and C. Diaz, "How Unique is Your .Onion? An Analysis of the Fingerprintability of Tor Onion Services," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 2021–2036, event-place: Dallas, Texas, USA.

[19] D. Rupprecht, K. Kohls, T. Holz, and C. Popper, "Breaking LTE on Layer Two," in *2019 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, May 2019, pp. 1121–1136.

[20] 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*. Miami Florida: ACM, May 2019, pp. 249–260.

[21] T. OConnor, R. Mohamed, M. Miettinen, W. Enck, B. Reaves, and A.-R. Sadeghi, "HomeSnitch: behavior transparency and control for smart home IoT devices," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. Miami Florida: ACM, May 2019, pp. 128–138.

[22] S. R. Hussain, M. Echeverria, O. Chowdhury, N. Li, and E. Bertino, "Privacy Attacks to the 4G and 5G Cellular Paging Protocols Using Side Channel Information," in *Proceedings 2019 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2019.

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

[24] B. Hong, S. Bae, and Y. Kim, "GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier," in *Proceedings 2018 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2018.

[25] A. Singla, S. R. Hussain, O. Chowdhury, E. Bertino, and N. Li, "Protecting the 4G and 5G Cellular Paging Protocols against Security and Privacy Attacks," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 1, pp. 126–142, Jan. 2020.

[26] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.-P. Seifert, "Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems," in *Proceedings 2016 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2016.

[27] J. Kölndorfer, N. Hopper, and Y. Kim, "Location Leaks on the GSM Air Interface," 2012, pp. 103–112.

[28] C. Sørseth, S. X. Zhou, S. F. Mjølsnes, and R. F. Olimid, "Experimental Analysis of Subscribers' Privacy Exposure by LTE Paging," *Wireless Personal Communications*, vol. 109, no. 1, pp. 675–693, Nov. 2019.

[29] R. P. Jover, "LTE security, protocol exploits and location tracking experimentation with low-cost software radio," *arXiv:1607.05171 [cs]*, Jul. 2016, arXiv: 1607.05171. [Online]. Available: http://arxiv.org/abs/1607.05171

[30] I. Bang, T. Kim, H. S. Jang, and D. K. Sung, "An Opportunistic Power Control Scheme for Mitigating User Location Tracking Attacks in Cellular Networks," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1131–1144, 2022.

[31] T. Byrd, V. Marojevic, and R. P. Jover, "CSAI: Open-Source Cellular Radio Access Network Security Analysis Instrument," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[32] N. Golde, K. Redon, and J.-P. Seifert, "Let Me Answer That for You: Exploiting Broadcast Information in Cellular Networks," in *22nd USENIX Security Symposium (USENIX Security 13)*. Washington, D.C.: USENIX Association, Aug. 2013, pp. 33–48.

[33] K. Fang and G. Yan, "Paging storm attacks against 4G/LTE networks from regional Android botnets: rationale, practicality, and implications," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. Linz Austria: ACM, Jul. 2020, pp. 295–305.

[34] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. La Porta, "On cellular botnets: measuring the impact of malicious devices on a cellular network core," in *Proceedings of the 16th ACM conference on Computer and communications security - CCS '09*. Chicago, Illinois, USA: ACM Press, 2009, p. 223.

[35] S. R. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino, "LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE," in *Proceedings 2018 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2018.

[36] S. Park, A. Shaik, R. Borgaonkar, A. Martin, and J.-P. Seifert, "White-Stingray: Evaluating IMSI Catchers Detection Applications," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*. Vancouver, BC: USENIX Association, Aug. 2017.

[37] A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, and E. Weippl, "IMSI-catch me if you can: IMSI-catcher-catchers," in *Proceedings of the 30th Annual Computer Security Applications Conference on - ACSAC '14*. New Orleans, Louisiana: ACM Press, 2014, pp. 246–255.

[38] S. F. Mjølsnes and R. F. Olimid, "Easy 4G/LTE IMSI Catchers for Non-Programmers," in *Computer Network Security*, J. Rak, J. Bay, I. Kotenko, L. Popyack, V. Skormin, and K. Szczypiorski, Eds. Cham: Springer International Publishing, 2017, pp. 235–246.

[39] D. Rupprecht, A. Dabrowski, T. Holz, E. Weippl, and C. Pöpper, "On Security Research Towards Future Mobile Network Generations," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2518–2542, 2018.

[40] M. Chlosta, D. Rupprecht, T. Holz, and C. Pöpper, "LTE security disabled: misconfiguration in commercial networks," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. Miami Florida: ACM, May 2019, pp. 261–266.

[41] M. T. Raza, F. M. Anwar, and S. Lu, "Exposing LTE Security Weaknesses at Protocol Inter-layer, and Inter-radio Interactions," in *Security and Privacy in Communication Networks*, X. Lin, A. Ghorbani, K. Ren, S. Zhu, and A. Zhang, Eds. Cham: Springer International Publishing, 2018, vol. 238, pp. 312–338, series Title: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.

[42] S. R. Hussain, M. Echeverria, A. Singla, O. Chowdhury, and E. Bertino, "Insecure connection bootstrapping in cellular networks: the root of all evil," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. Miami Florida: ACM, May 2019, pp. 1–11.

[43] M. Echeverria, Z. Ahmed, B. Wang, M. F. Arif, S. R. Hussain, and O. Chowdhury, "PHOENIX: Device-Centric Cellular Network Protocol Monitoring using Runtime Verification," p. 42.

[44] H. Khan, B. Dowling, and K. M. Martin, "Identity Confidentiality in 5G Mobile Telephony Systems," Tech. Rep. 876, 2018.

[45] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Call Me Maybe: Eavesdropping Encrypted LTE Calls With ReVoLTE," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 73–88.

[46] D. Rupprecht, K. Kohls, T. Holz, and C. Poepper, "IMP4GT: IMPersonation Attacks in 4G NeTworks," in *Proceedings 2020 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2020.

[47] D. Rupprecht, K. Jansen, and C. Pöpper, "Putting LTE Security Functions to the Test: A Framework to Evaluate Implementation Correctness," in

10th USENIX Workshop on Offensive Technologies (WOOT 16). Austin, TX: USENIX Association, Aug. 2016.

[48] Y. Chen, Y. Yao, X. Wang, D. Xu, C. Yue, X. Liu, K. Chen, H. Tang, and B. Liu, "Bookworm Game: Automatic Discovery of LTE Vulnerabilities Through Documentation Analysis," in *2021 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, May 2021, pp. 1197–1214.

[49] 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*. London United Kingdom: ACM, Nov. 2019, pp. 669–684.

[50] M. T. Raza, Y. Guo, S. Lu, and F. M. Anwar, "On Key Reinstallation Attacks over 4G LTE Control-Plane: Feasibility and Negative Impact," in *Annual Computer Security Applications Conference*. Virtual Event USA: ACM, Dec. 2021, pp. 877–886.

[51] R. N. Mitra, M. M. Kassem, J. Larrea, and M. K. Marina, "CUPS Hijacking in Mobile RAN Slicing: Modeling, Prototyping, and Analysis," in *2021 IEEE Conference on Communications and Network Security (CNS)*. Tempe, AZ, USA: IEEE, Oct. 2021, pp. 38–46.

APPENDIX

A. Acronyms

LTE Long Term Evolution
UE User Equipment
eNodeB Evolved Node B
EPC Evolved Packet Core
IMSI International Mobile Subscriber Identity
GUTI Globally Unique Temporary ID
RNTI Radio Network Temporary Identifier
SI-RNTI System Information RNTI
P-RNTI Paging RNTI
DCI Downlink Control Information
PRB Physical Resource Block
CCE Control Channel Element
MIB Master Information Block
SIB System Information Block
PDCCH Physical Downlink Control Channel
PDSCH Physical Downlink Shared Channel
PHICH Physical Hybrid ARQ Indicator Channel
SDR Software Defined Radio
RAN Radio Access Network
EUTRAN Evolved UMTS (Universal Mobile Telecommunications System) Terrestrial Radio Access Network
CRC Cyclic Redundancy Check
RLF Radio Link Failure
MSps Megasamples per second