Peter Ney*, Ian Smith*, Gabriel Cadamuro, and Tadayoshi Kohno

SeaGlass: Enabling City-Wide IMSI-Catcher Detection

Abstract: Cell-site simulators, also known as IMSI-catchers and stingrays, are used around the world by governments and criminals to track and eavesdrop on cell phones. Despite extensive public debate surrounding their use, few hard facts about them are available. For example, the richest sources of information on U.S. government cell-site simulator usage are from anonymous leaks, public records requests, and court proceedings. This lack of concrete information and the difficulty of independently obtaining such information hampers the public discussion. To address this deficiency, we build, deploy, and evaluate *SeaGlass*, a city-wide cell-site simulator detection network. SeaGlass consists of sensors that measure and upload data on the cellular environment to find the signatures of portable cell-site simulators. SeaGlass sensors are designed to be robust, low-maintenance, and deployable in vehicles for long durations. The data they generate is used to learn a city's network properties to find anomalies consistent with cell-site simulators. We installed SeaGlass sensors into 15 ridesharing vehicles across two cities, collecting two months of data in each city. Using this data, we evaluate the system and show how SeaGlass can be used to detect signatures of portable cell-site simulators. Finally, we evaluate our signature detection methods and discuss anomalies discovered in the data.

Keywords: cellular surveillance, cell-site simulator, IMSI-catcher, stingray, ridesharing, crowdsource

DOI 10.1515/popets-2017-0027
Received 2016-11-30; revised 2017-03-15; accepted 2017-03-16.

† The first two authors contributed equally to this work.
***Corresponding Author: Peter Ney:** University of Washington, E-mail: neyp@cs.washington.edu
***Corresponding Author: Ian Smith:** University of Washington, E-mail: imsmith@cs.washington.edu
Gabriel Cadamuro: University of Washington, E-mail: gabca@cs.washington.edu
Tadayoshi Kohno: University of Washington, E-mail: yoshi@cs.washington.edu

1 Introduction

Cell-site simulators, also known as IMSI-catchers or stingrays, act as rogue cellular base stations that can surveil cellphone locations and often eavesdrop on cellular communications. These devices are used extensively by governments and law enforcement, with devices coming in a wide range of capabilities. According to the Surveillance Catalogue, leaked by the Intercept, different models claim to intercept and record digital voice, geo-locate targets, and capture thousands of phones at once [27]. A series of open records requests and investigative journalism show that many U.S. police departments used them extensively, including Anaheim, CA; Baltimore, MD; Milwaukee, WI; New York, NY; and Tacoma, WA [8, 13, 20, 22, 33]. The U.S. Marshal's service has used airplane-mounted "DRT Box"' cell-site simulators to track fugitives since 2007 [3]. Other countries are using cell-site simulators as well, e.g., Ukraine's use of cell-site simulators to send a mass text to protesters during the Euromaidan protests [17].

Given their clear privacy implications, there is vigorous public discussion on their proper use and regulation. A key question and important topic among journalists, policy makers, and the legal community, is how often and in what context cell-site simulators are used, and whether they are being used responsibly. For the present, the public relies on data obtained from public records requests, court documents, and leaks to the press to understand government usage. We argue that the community — and those engaged in the policy debate surrounding cell-site simulator usage — would benefit from additional, independent sources of information on cell-site simulators.

To facilitate this goal, we developed *SeaGlass*, a system designed to detect cell-site simulators by longitudinally measuring and analyzing the cellular environment across any city. SeaGlass collects data about cellular networks using portable sensors that are placed in ridesharing vehicles. We designed sensors to be highly robust to failure, enabling long-term deployment in vehicles owned and operated by others. Each sensor collects data when the vehicle is powered on and uploads it to a cloud server for aggregation. We use this data to

develop methods for detecting anomalies or signatures that we expect from cell-site simulators.

To evaluate SeaGlass, and to iteratively refine our analysis pipeline, we deployed SeaGlass for two months in two cities: (1) Seattle, WA using nine drivers for eight weeks and (2) Milwaukee, WI using six drivers for eight weeks. Over the course of these deployments we collected 2.1 million unique cellular scans from locations across both cities. We then applied our analysis methods to our real data set and found that they detected base stations with anomalous signatures that might be expected from cell-site simulators. Our results suggest that if SeaGlass was deployed in a city where cell-site simulators are frequently used that they would be detected.

This paper contributes the following:

- We developed SeaGlass, a cost-effective, low-maintenance system to collect cellular environment data for detecting cell-site simulators on a city-wide scale using sensors in vehicles.
- We deployed SeaGlass in ridesharing vehicles in two U.S. cities, Seattle and Milwaukee, for nine and eight weeks respectively to evaluate our collection system.
- We designed methods to detect the identifying behaviors of cell-site simulators, and evaluated those methods using the data collected from the two SeaGlass deployments.

2 Basic Concepts

This section provides high-level background information and terminology on the GSM protocol and cell-site simulators. It also includes a discussion of signatures that cell-site simulators exhibit, which were used to develop our detection methods, and a list of publicly available sources of cellular network data.

2.1 The GSM Protocol

GSM, also known as 2G, is a cellular protocol first deployed in 1991 that remains in widespread use today. In the U.S. and Canada, it operates on 850 MHz and 1900 MHz, and in most other countries on 900 MHz and 1800 MHz. These bands contain uplink-downlink channel pairs, called Absolute Radio Frequency Channel Numbers (ARFCNs), on which phones communicate with Base Transceiver Stations (BTS), more generally known as base stations.

A network base station broadcasts its identifiers and other configuration properties to phones on a Broadcast Control Channel (BCCH). Phones generally choose to camp on the BTS in the network that has the highest received signal strength and best combination of BCCH properties. To register with the network, each GSM subscriber has a smart card, called a Subscriber Identity Module (SIM), which contains unique subscriber information, such as the International Mobile Subscriber Identity (IMSI). The IMSI is transmitted to base stations to identify the phone to the network. Because IMSIs are sent in the clear and can be linked to individual subscribers, phones typically negotiate a Temporary Mobile Subscriber Identity (TMSI) to attempt some privacy of the IMSI. The network can renegotiate a TMSI at any time, such as when a subscriber moves to a new geographical area — indicated by a BTS with a different Location Area Identity (LAI).

2.2 Cell-Site Simulators

Cell-site simulators have a variety of features. In their most basic form, they coax phones in the vicinity to reveal their IMSIs by imitating legitimate base stations. Once a target IMSI is retrieved, they may use directional antennas and received signal strengths gathered from multiple locations to localize a phone.

More advanced models give users the capabilities of a network provider. Many models offer active attacks like voice, SMS, and data traffic eavesdropping; injection; denial of service; cloning; and SMS spamming [27, 37]. GSM networks (2G) make these attacks possible because the network does not authenticate itself to phones. To exploit higher-level network protocols that support network authentication (3G and 4G LTE), some cell-site simulators take advantage of protocol vulnerabilities to downgrade to GSM before the network authenticates or jam on 4G/3G frequencies [5, 35]. In all protocol levels (2G, 3G, and 4G LTE), the IMSI is still transmitted as plaintext before network authentication, enabling some cell-site simulator functionality [32].

A recent leak of documents from the Harris Corporation (a cell-site simulator manufacturer known to sell to police departments) sheds some light on the strategies that common cell-site simulators use to capture phones [4]. Manuals for their RayFish product family, which includes the Stingray and Hailstorm models, indicate that these devices exploit a phone's cell reselection decision procedure by mimicking the weakest neighbor being advertised by a strong nearby base station. The

cell-site simulator then "transmits modified system information messages, including a modified Location Area Identifier (LAI) causing [a phone] to execute a location update," thereby revealing its IMSI, TMSI and IMEI [14]. We investigate this behavior and other ways cell-site simulators betray their use in the following section.

2.3 Signature Classes

To detect cell-site simulators, we first need to determine how they might betray themselves in measurable ways. We do not have access to commercial models for experimenting because their access is tightly controlled by manufacturers, so instead, we deduce their properties and behaviors from other sources such as document leaks and public records. Here we categorize these properties into broad classes of *signatures*, which we use as a basis for our detection methods described in §5.

Multi-location transmissions. Many cell-site simulators are portable in order to track or locate persons of interest [27]. To do so, they must relocate to transmit within the vicinity of a target. Portable cell-site simulators may also move while transmitting. For example, some models are handheld while others are attached to police cars or airplanes [3, 14, 15]. In all these cases, portable cell-site simulators will resemble a base station that changes location over time, which differs from the behavior of typical base stations. These multi-location transmissions are unavoidable for typical use cases and difficult to hide, so they can be used as a robust identifying feature.

Impermanence. Portable cell-site simulators are likely used for short durations and then powered off. This will resemble a base station that appears at a particular location for a short duration and then disappears.

Anomalous base station configurations. For improved efficacy capturing targeted phones or to minimize cellular network interference, cell-site simulators may transmit unusual broadcast control channel (BCCH) properties. Examples of these attributes include forcing phones to transmit more frequently than usual (low T3212) and preventing targeted phones from connecting to nearby neighbors (high cell reselect offset, empty neighbor lists) [10, 14, 30, 32].

Geographic inconsistency. Base stations advertise attributes determined by their network provider and geography. Unless cell-site simulators are carefully designed, they may appear out-of-place in this context. Two examples of these are out-of-place location area identities (LAI) and unexpected broadcast frequencies (ARFCNs) [14, 30].

Suspicious interaction with phones. A cell-site simulator interacts with a target phone differently than a typical BTS to achieve desired functionality, like capturing a phone's IMSI or eavesdropping on communications. These different interactions include using a weak cipher mode, communicating with the phone using 2G when it supports 4G, requesting an IMSI when a TMSI would be appropriate, advertising empty or unusual voice channels, or sending suspicious SMSs (e.g., silent SMS) [14, 30].

2.4 Crowdsourced Cellular Data Collection

To detect cell-site simulators, we want coverage of the cellular network across a city. In §3, we describe the SeaGlass vehicular sensors and how cellular data is collected. There are many services that collect cellular information to estimate location [2, 12, 19, 21, 23, 28, 31]. These services, like Google Location Service, use cellphone applications or other sensors to crowdsource radio signals from cell towers. Some services, like OpenCellID, also use this data to model the cellular network topology [21, 23, 24]. We discuss the applicability of these data sets in detecting cell-site simulators in §4.7, but we find that they are not sufficient to detect cell-site simulator use across a city.

Base station positions and network topology are known by the network carriers. This information could be used to more accurately flag cell-site simulator anomalies, however, carriers consider this information proprietary — the one exception in the U.S. is when antennas are on towers tallher than 200 feet and must be registered with the FCC [9]. Additionally, given the close involvement between telecommunications companies and government in many countries, we prefer to build a detection system that does not require network carrier cooperation.

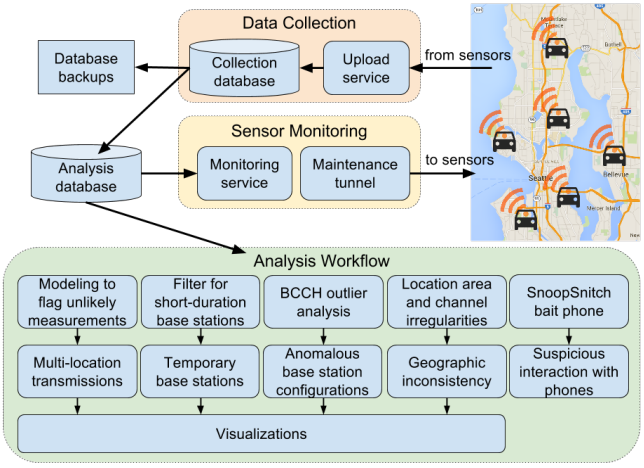

Fig. 1. SeaGlass system diagram. Sensors in vehicles scan the cellular environment and upload measurements for collection and analysis.

3 System Design

SeaGlass is designed to be a practical and cost-effective system to detect signatures of cell-site simulator in an urban environment. It gathers a wide and detailed view of the cellular network using sensors installed in vehicles; these sensors collect data continuously from different parts of the city as the vehicles drive around. It then models the largely static cellular network to flag anomalies in space, time, network parameters, and mobile-to-base station communications that may indicate the presence of cell-site simulators. SeaGlass has three main components: (1) vehicular cellular-scanning sensors that collect data as cars drive, (2) a cloud infrastructure that stores data and monitors sensor health, and (3) detection methods that identify signatures of cell-site simulators (Fig. 1).

This section describes each core component of SeaGlass in detail.

3.1 In-Vehicle Cellular Sensors

We designed sensors to record and upload relevant information about the cellular environment as they are driven around an urban area. They collect as much information about the GSM network as possible while remaining cost-effective, quick to prototype, and convenient for volunteers to maintain in their vehicles. Fig. 2 shows a sample sensor with the contents removed. Each sensor contains the following core components:

- **GPS**: Records time, 3D position, speed, heading, and accuracy.
- **GSM modem**: Scans every ten seconds, recording a list of all base station frequencies, received signal strengths, bit error rates, and BCCH properties detected at each location.
- **Bait phone**: An Android phone running SnoopSnitch, which collects network packet captures, suspicious events, and the list of 4G, 3G, and 2G base stations on which the phone camps.
- **Raspberry Pi**: Continuously caches and uploads aggregated data from the modem, GPS, and bait phone.
- **Hotspot**: Provides Internet connectivity to upload data and allows a reverse shell into the Raspberry Pi for remote debugging.

The GSM modem collects the frequencies of all detected voice and BCCH channels and additional properties transmitted on the BCCH, such as the unique base station identifiers (MCC, MNC, LAC, BSIC, CID), the BA list, the ARFCN list, cell status, GPRS parameters, and other properties. The modem gives us much higher visibility into the GSM bands than any smartphone app can because the smartphone basebands generally only report the strength of the BTS currently camped on, and rarely the strength of the neighbors. With an external antenna attached, the GSM modem can report dozens of BCCHs and hundreds of voice channels in a single scan. The modem also reports many different BCCH properties for each BTS, which smartphone operating systems typically cannot access. The choice of modem over commodity smartphone results in longer detection ranges and vastly richer data collection ability.

The sensors are installed in the trunks of vehicles. To ensure good reception of the modem antenna, GPS, and bait phone, they are mounted on or near windows. The sensor is powered by the DC 12 V cigarette lighter outlet via an extension cord. When the outlet provides power, the sensor automatically boots and starts collecting data without manual intervention. Both modem and GPS data are written to a local persistent database running on the Raspberry Pi. When Internet connectivity is available via the hotspot, the cached data in the local database and the SnoopSnitch logs (pulled from the bait phone) are uploaded to a cloud collection database and deleted from local storage.

To make SeaGlass cost-effective, the underlying equipment must be relatively inexpensive. The one-time cost for each sensor's components was $502 (Appendix A for parts and costs). Each sensor also requires

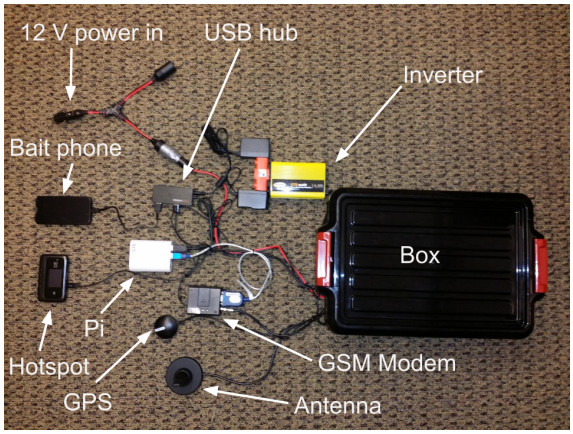

Fig. 2. A vehicular sensor with contents removed and labeled. Components are packed into the box and installed into a vehicle's trunk, with antennas placed on or near windows.

a data plan for the hotspot, at about $10/mo. Thus, the equipment cost for 10 sensors is $5,020, plus an annual operating cost for 10 data plans is about $1,200 per year. The total of $6,220 for ten sensors the first year is a fraction of the cost of a single typical cell surveyor device, which often start over $100,000 [25].

Given the present cell-site simulator landscape, we chose to limit this study to the GSM network. While 4G-capable cell-site simulators exist (e.g., Harris Corporation Hailstorm), it is unclear how many of them are used in practice due to their newness and expense [11]. In contrast, GSM cell-site simulators have been used much longer and can still be effective against 3G and 4G phones by jamming and downgrading attacks [35]. By focusing on a single protocol, we decreased the cost and increased the simplicity of the sensors, allowing us to more quickly build a deployable system to advance our understanding cell-site simulator detection. However, the system design and data collection is general and can be applied to all current cellular protocols (2G, 3G, and 4G).

3.2 Cloud Infrastructure

The second component of SeaGlass, its cloud infrastructure, stores uploaded sensor data, monitors problems with live sensors, and analyzes results.

The vehicle's sensors upload data every few minutes to a REST API, where data is sanitized, parsed, and stored in a persistent MongoDB database. For robustness, this database is backed up daily, and all analysis uses an offline PostgreSQL database that is populated periodically from the live MongoDB database.

To keep the sensors highly available, we wrote monitoring software to detect problems on sensors. A reverse SSH tunnel allows us to access sensors remotely, which is useful to fix software problems or push updates.

3.3 Signature Detection Methods

Detection methods comprise the final component of SeaGlass, and include a set of analyses, modeling techniques, and visualizations that find signatures of cell-site simulators in the collected data. Our prototype deployments collected a significant amount of sensor data (183 MB of raw scan data per day, averaging about 35,000 scans per day) which makes it especially important to automate the process of finding rare signatures.

We describe and evaluate these techniques in detail in §5. Below we summarize our broad analysis categories. See Table 1 for the corresponding signatures each analysis is designed to detect.

- **Modeling base stations to flag unlikely measurements.** By collecting multiple measurements of each base station from many locations and using theoretical fading estimates of cellular signals, we model base station locations and estimate parameters such as power and height. We use these models to flag measurements that are unlikely from a stationary BTS, indicating they may originate from multiple locations.
- **Filtering for short-duration base stations.** We use short-lived BCCH broadcasts as a sign that a base stations is impermanent.
- **Identifying BCCH outliers.** We collect the full set of attributes that are transmitted over the broadcast control channel and look for outliers and specific attributes known to be associated with cell-site simulators.
- **Finding location area and channel irregularities.** We learn the location area codes and broadcast channels that are used by each carrier in a geographic area and identify base stations that do not fit.
- **Using a bait phone.** Each sensor is equipped with an Android phone that runs SnoopSnitch — an app that logs suspicious connection-level events — to act as bait for a cell-site simulator. We can use the SnoopSnitch logs to determine if there have been any suspicious events seen by the phone.

Signature	Detection Method	Data Required	Minimum # of Measurements
Multi-location transmission	Model base stations to flag unlikely measurements (§5.1)	BTS identifiers; received signal strength; GPS	10
Impermanence	Filter for short-duration base stations (§5.2)	BTS identifiers	3
Anomalous base station configurations	Identify BCCH outliers (§5.3)	BCCH properties (e.g., channel, cell status, GPRS properties, etc)	1
Geographic inconsistency	Find location area and channel irregularities (§5.4)	LAC; BCCH frequency; GPS	1
Suspicious interaction with phones	Use a bait phone (§5.5)	Phones running SnoopSnitch	1

Table 1. Cell-site simulator signatures and their corresponding detection methods, the data collected for detection, and the number of measurements required for detection.

4 Deployments

This section describes and evaluates the deployments of SeaGlass for approximately two months in Seattle, WA and Milwaukee, WI. We use the collected data to understand key properties of the system, including data quantity and the impact of different deployment conditions. We also use this data to generate and evaluate signature detection methods, described in §5.

4.1 Cities

Our choice of trial cites was determined by many factors, including evidence of regular cell-site simulator use, the diversity of local regulations between the trial cities, and different urban conditions.

Milwaukee, WI. Information from a 2015 public records request reported that the Milwaukee police department used cell-site simulators an average of 10 times per month from 2011-2015 [36]. The report details each cell-site simulator event, including specific dates and times, districts deployed, the crime being investigated, and, in some cases, specific locations. If similar records are released in the future, they would be useful to corroborate our findings.

Seattle, WA. In 2015, Washington became one of the first states to pass legislation that explicitly prohibits the use of cell-site simulators without a warrant [34]. While this does not ban them, it may reduce their usage by authorities. This legislation lets us compare cell-site simulator use between different regulatory regimes. We also chose Seattle due to its proximity to Tacoma, WA, an epicenter of cell-site simulator policy discussion due to the discovery that local police were using them without explicit judicial oversight [7].

4.2 Driver Information

We recruited ridesharing drivers to collect data in these cities because they drive for many hours and cover diverse areas. To qualify, they had to drive a minimum of 20 hours per week and were compensated $25 per week for their time and $100 when they successfully returned the equipment.

Sensors were installed in nine ridesharing cars for nine weeks (March 17, 2016 - May, 17 2016) in Seattle, and in six cars for eight weeks (March 23, 2016 - May 17, 2016) in Milwaukee. One week in mid-study (from April 5 - April 12) had only intermittent collection because we switched from a roof-mounted antenna to a window-mounted one that was more convenient for our drivers.

Extensive discussions with our university Institutional Review Board confirmed that this study was not considered human subjects research. However, we ensured that our drivers understood the purpose and risk of gathering data for our study — in particular, that the sensor collects location data about the vehicle whenever it is running. We adopted some human research practices including requiring that drivers read and sign a consent form to participate, and anonymizing and otherwise protecting all collected data.

4.3 Coverage, Density, and Diversity

Over the course of the two deployments, we collected data from over 2.1 million cellular scans (1.4 million in Seattle and 600,000 in Milwaukee). In terms of city land area, cellular scans were made in 215/217 km^2 in Seattle

and in 241/249 km^2 in Milwaukee. Fig. 3 shows a density plot of the number of cellular scans made in each square km. Scan density varied widely depending on the region in each city. In dense population areas, especially around downtowns, there were more than 10,000 scans/km^2, while low-density outskirts had as few as 100 scans/km^2.

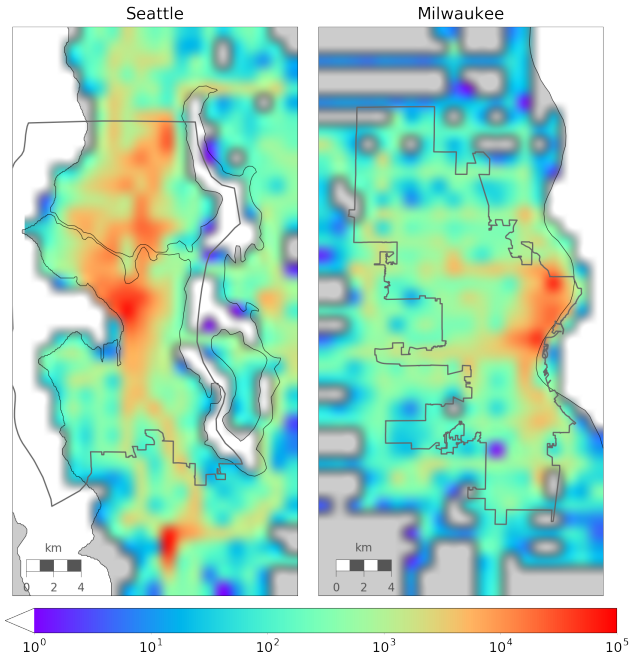

Fig. 3. Seattle (left) and Milwaukee (right) driver coverage maps, with city limits boundaries drawn in black. Bins are 1 km across. The color bar (log-scale) shows the number of measurements per bin during our 9-week duration in Seattle and 8-week duration in Milwaukee.

Another measure of scan coverage is the number of measurements collected for each BTS. Our choice of modeling algorithm (which we use in Section 5.1 to discover the special case of mimicking cell-site simulators) works best with at least 10 measurements per BTS. In general, more measurements of a BTS lead to a better fit with our models. In our data, we find the number of measurements per base station is heavily skewed. A few base stations have over 10,000 measurements, while the average is 321 in Seattle and 145 in Milwaukee. However, 83.7% of the base stations we model have at least 10 measurements, and we expect that longer deployments will increase the number of measurements per BTS.

In addition to high coverage density, we seek a high *coverage diversity*, meaning that for every BTS we want measurements from a variety of locations. High coverage diversity improves our modeling results and ensures the modeling is more robust to local effects, such as shadowing (attenuation caused by obstructing objects).

To measure location diversity, for each base station, we counted the number of unique 100m x 100m geographical bins from which a measurement was taken. In Seattle, the mean was 117 unique bins per BTS and the median was 83 unique bins per BTS, while in Milwaukee, the mean was 83 and median 44. These results proved sufficiently large for our modeling requirements and much larger than what could be expected from stationary sensors, which can individually only measure from a single position per BTS.

4.4 Canvassing Rate

Many signature detection methods depend on a well measured and stable view of the underlying cellular network. The faster SeaGlass can sufficiently canvass a city, the less time it needs to operate before becoming an effective detection system. One way to measure the canvassing rate is to count the number of new base stations encountered by sensors each day. In both cities, 80% of the base stations encountered during the deployment were canvassed within about the first 10 days (10 days for Seattle and 12 days for Milwaukee).

The data also highlights some interesting differences between Seattle and Milwaukee. Both cities have similar area, but Milwaukee requires fewer base stations (609 BTSs for Milwaukee compared to 1411 for Seattle). This is likely due to its lower population density than Seattle and the longer range of base stations over its flatter terrain.

4.5 Sensor Receive Range

Another important property is the receive range of the sensors. The greater the receive range, the higher the detection radius for each sensor, which increases the likelihood that a sensor is in range of a cell-site simulator transmission.

Receive range shows high variability because of its dependence on many factors, including surrounding topography, density of buildings, directionality of base stations, and weather. Also, it depends on the transmission power of the BTS, which can vary substantially among cell-site simulators (e.g., handheld vs aerial models).

To approximate sensor receive range, we use Google Location Services to roughly estimate the location of a base station. Centered around this location, the receive

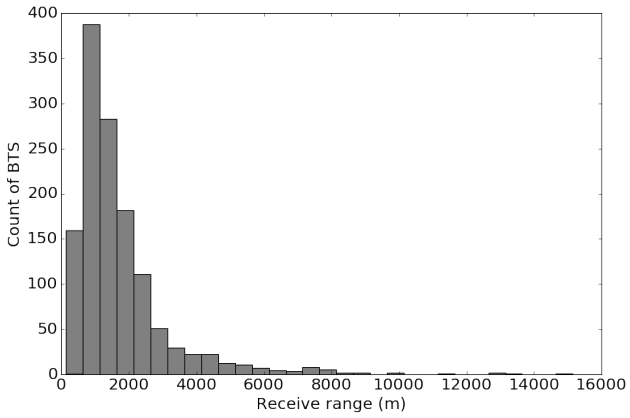

Fig. 4. Plot of distances that a sensor can reliably detect a base station in Milwaukee and Seattle (mean: 1731m, median: 1303m).

range is the distance of the 90th-percentile furthest measurement (the 90th-percentile reduces the impact of extreme outliers).

Our data shows that the median distance our sensor can reliably detect a base station is 1.3 km, and the average is 1.7 km (Fig. 4). If we include all measurements, this range varies from 150 m for low-power base stations in dense downtown areas to more than 150 km in open areas at high elevations or across water (such as Lake Michigan). We do not expect this to be a perfect estimate, but it provides a useful approximation that will be helpful in our subsequent analysis.

4.6 Detection Probability

Because ground truth of cell-site simulator use is not publicly known, we use a simulation to estimate the probability that SeaGlass could have detected a cell-site simulator if one were used at various times and places. To detect a cell-site simulator, a SeaGlass sensor must be within range of it while it is transmitting. Therefore, our detection ability is bounded by the frequency that a cell-site simulator is operating within range of a sensor. It also depends on deployment factors, like sensor range, density of coverage, and number of sensors, and also how frequently, for how long, and in what part of the city cell-site simulators are used.

To give a more quantitative measure of our detection ability, we simulate cell-site simulator events for varying durations in Seattle and Milwaukee over the course of the deployments. Then we measure the probability that a vehicular sensor would have been within range of that event, and thus, whether the event could have been measured. We begin by randomly choosing 100,000 (time, location) event tuples, each simulating a cell-site simulator transmission at a location (contained within the Seattle and Milwaukee city limits) for a specific amount of time (ranging from 1-24 hours).

A cell-site simulator can be detected only if it has been measured at least once, so we can use the percentage of simulated transmission within range of one of our vehicular sensors as a lower bound on our detection probability. We consider a sensor to be within range of a simulated event if it was within 1303 m of the transmission (the median sensor receive range).

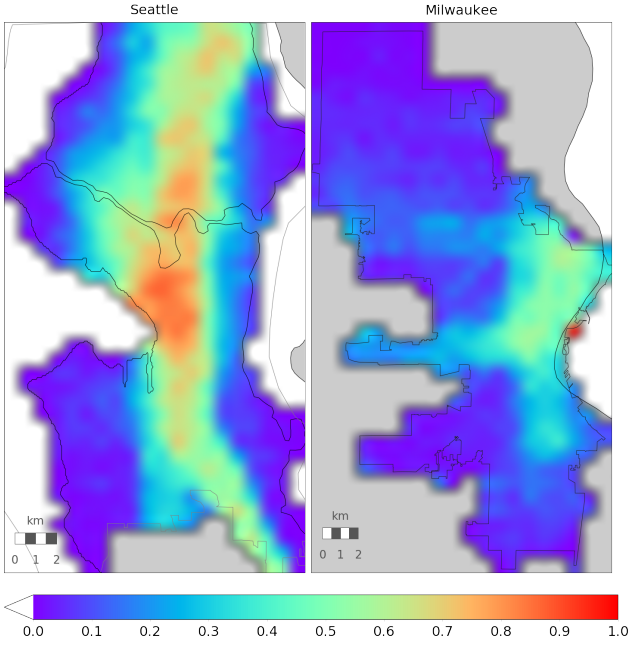

Fig. 5. Heatmaps of Seattle (left) and Milwaukee (right) showing the probability that a cell-site simulator would have been collected in the data if it were deployed at locations for a 2-hour duration with a detection radius of 1303 meters. These heatmaps correspond to the points plotted on the curves in Fig. 6.

Fig. 5 shows two heatmaps with the detection probabilities that a sensor was within range of a simulated event, assuming that a cell-site simulator was operational for two hours. These results are expected: in the densest parts of both cities, near downtowns, we have the highest detection probability, while in the outskirts it is much lower. Note, that we chose to draw points randomly in space and time, but in reality cell-site simulators may be used more frequently in high-density areas or at particular times (e.g., normal working hours), which may correlate better with our coverage.

To evaluate how different deployment conditions, like the number of vehicular sensors, affect our ability

to detect cell-site simulators, we simulated a different number sensors (6, 9, and 18 sensors in Seattle and 4, 6, and 12 in Milwaukee). To simulate fewer sensors, we could not randomly remove drivers until we reached the desired number because they do not drive uniformly. Instead, we gather all subsets of drivers with the desired number of drivers and then compute the total detection probability as the average detection probabilty from each subset. For example, if there were 10 drivers but we wanted to simulate 8, we would remove all possible combinations of two drivers and then average the detection probabilities from each of those combinations. To simulate an increase in the number of drivers, we doubled the number of sensors by dividing the deployment duration in half and rewriting all measurements in the second half to the first.

We also simulated how different run times of cell-site simulators (1-24 hours) affect the detection probability. Further, because some analyses require more than a single measurement of a cell-site simulator to be effective (e.g., detecting temporary or short lived transmission), we compute the detection probability when both one or ten measurements of a given cell-site simulator event is needed. See Fig. 6 for the detection probabilities under these different deployment conditions.

In Seattle, when the cell-site simulator duration is longer than 12 hours the probability of detection is fairly high — over 68% with one measurement and 38% with ten — and quickly plateaus as the duration increases. When the duration is short, there is a much lower detection probability.

The benefit of additional sensors depends mostly on the cell-site simulator duration. If it is live for short periods, then doubling the number of sensors can increase the detection probability by a factor 1.5-2.0x — for example, with a one hour duration and a only a single measurement required, the detection probability increases from 24% to 36% in Seattle (a factor of 1.5) and in Milwaukee from 9% to 16% (a factor of 1.8).

As more measurements are needed, additional sensors have a bigger impact. With 10-measurements needed, doubling the number of drivers raises detection probabilities by 2.0 for the 1-hour durations in Seattle and Milwaukee. However, as the duration becomes longer the increased benefit of more sensors becomes marginal. With a 24-hour duration, doubling the drivers increases the detection by a factor of only 1.1 - 1.5.

The number of sensors is an important deployment condition that should be selected based on many factors, like the desired detection probability, expected duration of cell-site simulators, and number of measurements required of each event. This analysis shows that if one is looking for very short transmission (1 hour), then the detection probability will be fairly low, and many cell-site simulator events will be missed, but each additional driver will make a large improvement. If the expected transmissions are long (greater than 12 hours) then the detection probability should be high, even with a low number of sensors. As more measurements of each event are required, the detection probability will be lower and will improve with longer cell-site simulator durations.

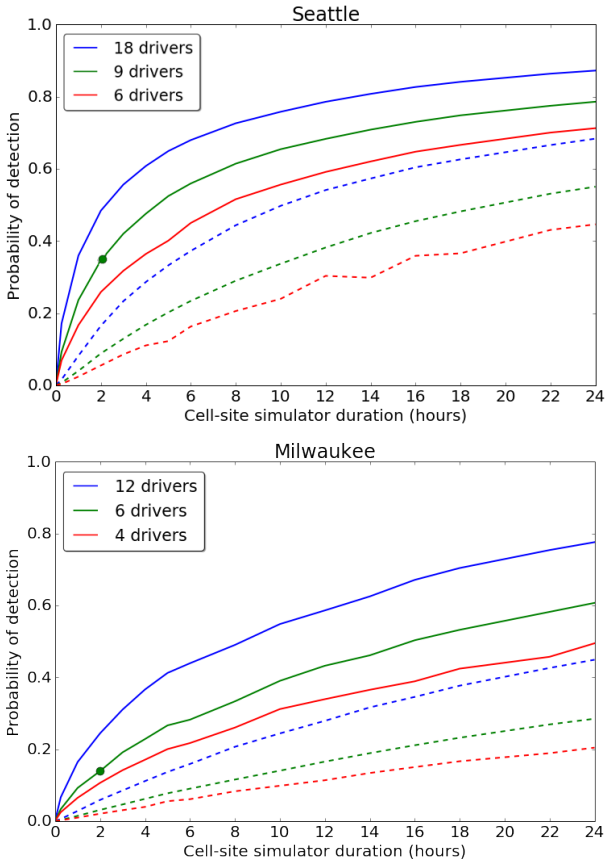

Fig. 6. Plot of detection probabilities for varying durations of cell-site simulators and number of drivers in the Seattle (top) and Milwaukee (bottom) data sets. Cell-site simulator detection radius parameter set to 1303 m. The solid lines are for a minimum of 1 measurement, and dashed is for minimum of 10. The green points represent the parameters used to create the heatmap in Fig. 5.

4.7 Comparison with Other Data Sources

In §2.4 we described the publicly available crowdsourced datasets, like Mozila Location Services and OpenCellID, that record BTS broadcast information collected from cell phones. Could this data be used directly to detect

cell-site simulators or as a supplement to the data collected by SeaGlass?

Since it is likely that cell-site simulators mimic nearby BTS identifiers, aggregated measurements will not be useful by themselves, and so it is important to have non-aggregated, raw measurements of each base station (as will be shown when we describe our signature detection methods in §5). Currently, OpenCellID is the only large, public dataset that contains raw measurement information. To effectively detect short-lived and portable cell-site simulators, this data needs to have high coverage and density. However, when comparing the OpenCellID dataset with SeaGlass, it is clear that their dataset is not sufficient. For example, in the Seattle city limits, OpenCellID contains 6742 measurements of 763 distinct GSM base stations from 12/03/2011 - 11/03/2016, while SeaGlass collected 660,000 measurements of 1411 GSM base stations in Seattle city limits in just 9 weeks. The OpenCellID data is more dense in Milwaukee, but it is still far less than SeaGlass — there were 20% fewer measurements and 59% fewer base stations detected over a time window 14 times larger as the SeaGlass Milwaukee deployment.

The OpenCellID data is further limited because smartphone basebands do not let applications broadly scan for base stations and only report the serving cell and its neighbors, so they may miss collecting weaker or unexpected base stations not present in the neighbor's list. OpenCellID also reports only a subset of the BCCH properties and excludes 19 of the extended BCCH properties, some of which are relevant to cell-site simulator detection (see §5.3). These limitations of phones are a fundamental, technical limitation to any smartphone-based data collection platform. The BTS location estimates provided by many of these location services are, however, a useful independent source of information to validate our collection methods and modeling.

5 Signature Detection

SeaGlass seeks to detect rare signatures, but the size and complexity of the data make finding infrequent events challenging. This challenge is exacerbated because the precise capabilities of cell-site simulators are not publicly available and may vary across models.

To deal with this detection problem, we take a "defense in depth" approach, by analyzing the data over a wide variety of signatures. If anomalies are found, especially across multiple dimensions, our confidence that the anomaly is a real cell-site simulator event increases. Looking for a broad class of signatures also makes detection more robust to future defenses against monitoring. Over time, as we collect more data, we will continue to hone and improve these methods.

This section examines the details and results of our detection methods: modeling for low-likelihood measurements (§5.1), filtering for temporary base stations (§5.2), identifying BCCH outliers (§5.3), finding location area and channel irregularities (§5.4), and using bait phones (§5.5).

5.1 Modeling for Low-Likelihood Measurements

Many cell-site simulators, like the Harris Corporation StingRay and Boeing DRT Box, are designed to be portable and move while operating [3, 16]. Such devices necessarily transmit at different locations over time, in contrast to a stationary BTS. Thus, detecting transmissions from different positions can act as a proxy for movement and provides a robust signature that cannot be easily disguised.

Depending on how a cell-site simulator is configured, it may advertise a constant BTS identifier as it moves around, or it may spoof the identifier of an existing BTS. In both cases the same BTS identifier is advertised from different locations over time, which is unexpected in a normally-stationary BTS. These multi-location transmissions can be detected when the received signals are measured in new or unexpected locations.

To quantify the unexpectedness of received signal strengths, we model each BTS to generate a statistical likelihood for each measurement. The more statistically unlikely the measurement, the more poorly it was fit into the model, indicating a relevant outlier. If a cell-site simulator does not spoof existing BTS identifiers or uses a constant identifier, then every time it operates it will appear as a short-lived base station and can be detected by other methods (see §5.2).

Base Station Modeling Method. We cannot directly learn physical base station properties, such as height, transmit strength, and position, with SeaGlass sensors. However, using the received signals the sensors collect, we can estimate these parameters by modeling base stations in the collected data. These physical parameter estimates can then be used to determine the likelihood of each measurement.

To model a BTS, we start with an urban cell and fading model to approximate how signals fall-off as they

propagate in a city and assume that typical base stations are stationary. Using the received signal strengths and bit error rates from the BCCH broadcasts, we can estimate BTS parameters (transmission power, antenna height, and location) that best fits the data. With these parameter estimates, the statistical likelihood of each measurement is determined, and those measurements with a low likelihood are marked as suspicious and flagged for further inspection. For modeling to be useful, there must be enough measurements of a BTS, so we only attempt to model base stations with at least 10 measurements collected. Base stations with fewer measurements would either be flagged as a temporary base station (§5.2), or would not have been measured frequently enough to collect a baseline for distinguishing anomalous signal strengths. We focus here on our overall infrastructure and defer further details of our base station modeling algorithm to Appendix B.

Inspection of Real Data. To identify base stations with poorly fit measurements, we compute the likelihood of all measurements and filter for base stations with measurements that have likelihoods at least three standard deviations less than the mean. In Seattle, this flags 13 base stations out of 1137, only 1.1% of the total, which is a manageable number to manually inspect.

We manually inspected the 13 flagged base stations in Seattle to see if they exhibited the behavior that would be expected from multi-location transmissions. Five of the 13 had low-likelihood measurements that were distant from a main cluster (example in Fig. 7). This is what would be expected when cell-site simulator mimics another base station and is detected with a single measurement. Another two base stations have a low-likelihood because there is a strong received signal inside a dense cluster of weak measurements, which could occur with a strongly transmitting cell-site simulator. However, we think it is unlikely that the anomalies in both of these categories are due to an actual cell-site simulator because there is no corroborating evidence from other signature classes. These measurements could have been caused by signals traveling across water (which violate the urban fading assumption in the model), cell-on-wheels, or multipath effects.

The six remaining base stations were clearly false positives: three had a few weak measurements next to many strong ones, probably due to interference or shadowing, and the other three base stations exhibited irregular measurement distributions and were not well modeled.

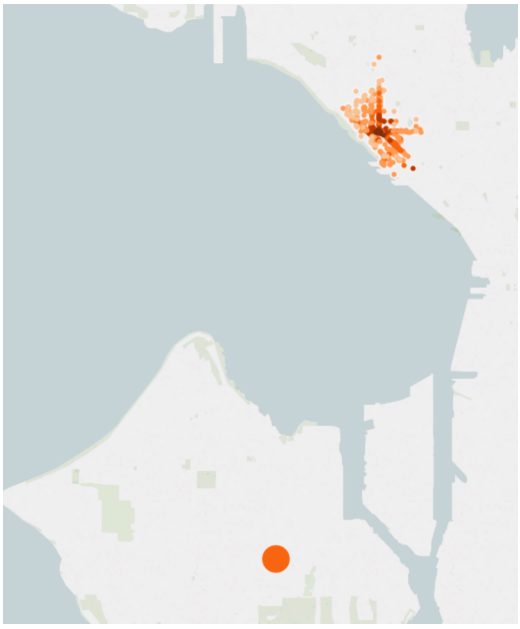

Fig. 7. Example BTS found outside the 3-std cutoff with an unlikely measurement 5 km away. Color represents the received signal strength of the measurement (darker is higher strength) and point size represents the likelihood computed by the model (larger is lower likelihood). All but one of the measurements of this base station are clustered downtown, and the other is measured 5 km away, with RSSI of -70 dBm and likelihood of -260.

Again, we think it is unlikely that the flagged anomalies came from cell-site simulators because there was no corroborating evidence from our other detection methods. However, our results give us confidence that this method is effective at detecting base stations with measurement patterns similar to portable cell-site simulators. The number of base stations flagged was manageable to manually inspect and identified relevant base stations better than half of the time. The same collection and detection methods also generalize to 3G and 4G protocols, allowing detection of newer cell-site simulators with modems that can scan those protocols.

Modeling Simulations. The leaked RayFish product manuals indicate that a common tactic for cell-site simulators is to mimic the identifiers of a nearby BTS — specifically, the identiers of the weakest neighbor of the BTS with the strongest signal. Given this possibility, we want to systematically evaluate our ability to detect a cell-site simulator that is mimicking another BTS under varying conditions, and in particular, determine if it is possible for a cell-site simulator to transmit close enough to the actual BTS that it effictively "hides" within its transmissions. To answer this, we simulate cell-site sim-

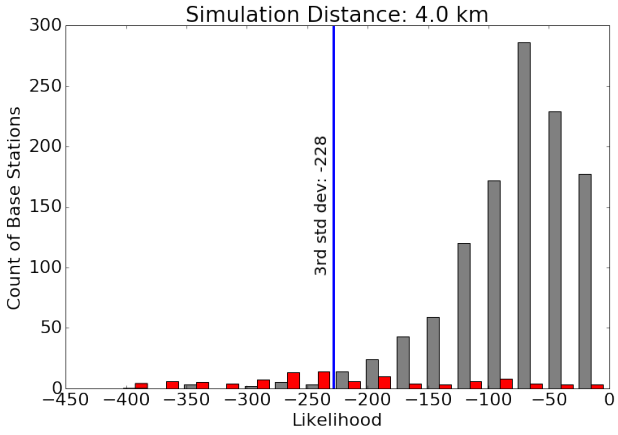

Fig. 8. The count of base stations with their lowest-likelihood point. Base stations from the data set in gray, and base stations with a point injected at 4 km away with a signal strength of -49 dB in red.

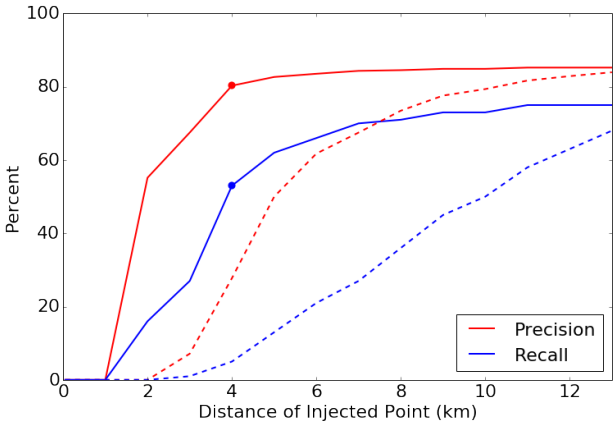

Fig. 9. Recall and precision of unlikely measurement detection simulations showing higher recall and precision for points injected farther away from the BTS and with higher signal strengths (solid is -64 dB, dotted is -49 dB). Points at 4 km indicate the distance and signal strength referenced in Fig. 8.

ulator measurements that are recorded at varying distances away from the mimicked BTS and analyze their detectability.

To begin simulating, we gathered 100 random base stations from the Seattle data that have at least one measurement taken in the city limits and at least ten measurements taken in total (so we can model them). To represent the case when a single measurement is made of a mimicking cell-site simulator, we injected 100 measurements into the data, one for each of the random base stations, and then modeled the base stations to compute their lowest likelihood measurement. We repeat this with injected measurements of varying distances away from the BTS (using Google Location Services to provide the approximate BTS location), ranging from 50 meters to 13 kilometers, and with signal strengths of either -64 dB or -49 dB, corresponding to one and two standard deviations stronger than the mean received signal strength.

Fig. 8 shows results of an example simulation with the lowest measurement likelihood for each BTS in Seattle (gray) and the 100 injected base stations (red), using injected measurements that are 4km away with a received signal strength of -49 dB.

Over 80% (53 / 66) of the base stations flagged as anomalous (with likelihoods below the 3-standard-deviation cutoff) are the base stations with injected points. However, 47% of the injected base stations were not considered anomalies. Since ground truth is not available we do not know the true accuracy of this method, but if we consider the injected base stations as true positives, then the false positive rate is 20% while the false negative rate is 4%.

Fig. 9 shows how recall and precision vary by distance and received signal strength. Assuming that the injected base stations are the only true positives, recall is the fraction of injected base stations that have a measurement likelihood below the 3-standard-deviation cutoff, and precision is the fraction of flagged base stations that we injected.

The recall and precision curves show that base stations injected with a nearby point (1 km or closer) were not detectable (zero recall and precision). However, as the injected point gets farther away, the recall and precision improve. The stronger the signal strength, the closer the distances that can be detected. With the -49 dB injected measurements (solid line), there is a sharp improvement around 4 km where both the recall and precision greatly improve and then quickly plateau to around 80% and 70%, respectively. With -64 dB injections (dotted line) at 6 km, the precision improves similarly, but the recall grows more gradually.

This analysis shows that it may be possible for a cell-site simulator to mimic a nearby BTS (0-2 km) and avoid detection by this method, but as it moves further away from the mimicked BTS it quickly becomes detectable. In practice, they may not transmit too close to the true BTS because it could cause interference, which may explain why they mimic the weakest neighbor of the strongest BTS. To avoid interference they could transmit on a different channel than the mimicked BTS, but that would be easily detectable by our channel analysis (described in 5.4). Further, because cell-site simulators have been operating with little adversarial detection pressure, they may be optimized for capture rate,

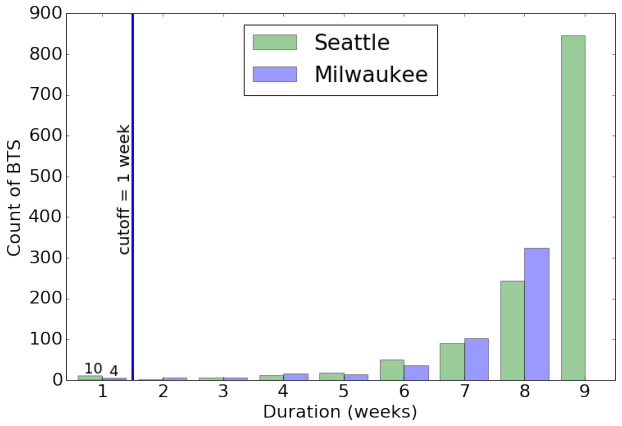

Fig. 10. Histogram showing the measured durations of base stations in Seattle (green) and Milwaukee (blue).

power, minimal interference or other features at the expense of easier detection.

5.2 Filtering for Short-Duration Base Stations

Another detectable signature of portable cell-site simulators is that they may transmit for short durations, on the order of minutes to days. For cell-site simulator models that do not have the capability to mimic base stations, then we can find short-lived base stations by filtering for those that broadcast only within a narrow time window. For this to be a useful filter, there must not be too many false positives. If the network base stations are not persistent or undergo frequent maintenance, then legitimate short-duration base stations might be common, but we find they are not. Our data shows that base stations typically continue to transmit from a single location for multiple weeks or longer, so we consider a BTS short-lived if all of its broadcasts are measured within a single one-week (7-day) window.

Fig. 10 shows the duration of all base stations with measurements in Seattle and Milwaukee that have at least three measurements — we require three measurements because any fewer and the BTS would be trivially short-lived (i.e., the duration of a BTS with a single measurement must be within one week). This leaves 1274 base stations Seattle and 506 in Milwaukee. The majority of them were live for the full duration of the deployment (9-weeks in Seattle and 8-weeks in Milwaukee), but 10 had 1-week or shorter durations in Seattle, as did four in Milwaukee. This is consistent with earlier analysis which showed that the underlying network is mostly stable, and our coverage is high in both cities.

Nine of the ten flagged base stations in Seattle were edge cases that are artifacts of the deployment duration: they either turned off in the first week or became live in the final week. These are unlikely to actually be temporary because six that turned off were measured regularly in prior test deployments and the three that turned on were regularly measured in tight clusters and their cell IDs indicate that they are three sectors on the same physical tower. This shows that there is a small amount of turnover of base stations (approximately 5 per week), but that once turned on, they are live for many weeks. The remaining outlier is located across Puget Sound and was flagged because it is rarely detected.

In Milwaukee, three of the four base stations were rarely measured and are located over 100 km away across Lake Michigan. The fourth base station is more interesting, but is likely a false positive. There were five measurements scattered across 40 km that were taken in the last week of the deployment. A base station with that range normally shows up in hundreds of scans. Further, it was transmitting on a unique channel for Milwaukee. However, all of the measurements were extremely weak (-100 dB to -105 dB) and along the same vector. Therefore, the most likely explanation is that there was a far away and strongly transmitting base station, that was directional along that vector. The unusual ARFCN is probably caused by a different ARFCN allocation in the area it is transmitting. Regardless, this hypothesis would easily be validated if the deployment continued longer.

These results show that filtering for short-lived BCCH broadcasts leaves a manageable number of base stations to manually inspect, and if a cell-site simulator was short-lived, then it would be flagged. There are situations where filtering for short-lived BCCH broadcasts with this method will not detect all base stations that temporarily transmit. For example, if a cell-site simulator is used repeatedly for short durations, and advertises the same identifier each time, then the broadcasts for that BTS would cover a wide time interval, and thus not be classified as short-lived. Similarly, if a cell-site simulator was configured to mimic the identifier of a BTS, then its broadcasts would hide in the broadcasts of the mimicked BTS, which is not short-lived. However, both of these cases have a BTS that is transmitting in multiple locations, and may be flagged by the modeling method described in the previous section.

5.3 BCCH Outlier Detection

In addition to physical-layer signatures like location and time, we look also for anomalies in broadcast control channel properties. The sensors collect 22 different (non-identifier) BCCH properties, which provide ample opportunity to identify cell-site simulators. There are two reasons that the BCCH properties of cell-site simulators may be different than typical base stations: (1) to cause useful behavior in a phone (e.g., cause a phone to update its location), or (2) because they do not perfectly mimic the properties of the network they are spoofing.

Each different network has its own idiosyncratic properties that vary by city and are highly consistent within providers. Cell-site simulators also advertise a mobile network code (MNC) to attract phones of that network. If the cell-site simulator is not configured to camouflage itself based on the specific network and location, these proprieties would likely differ, betraying their use.

To see whether it is possible to identify cell-site simulators with BCCH outliers, we identified correlations between network providers in Seattle and their BCCH properties and found that 14 of them are highly correlated with the network provider. For example, among 500,000 total measurements, the ALPHA property is 8 for all T-Mobile base stations and 10 for all AT&T. These 14 properties can be used to detect cell-site simulators which have not accurately copied the configuration of the mimicked network. (There was a technical issue with the GSM modem we used that caused it to report incomplete sets of BCCH properties 40% of the time, but we do not believe this impacts our BCCH correlation analysis other than reducing the total number of measurements.) We defer to Table 3 in Appendix C for additional statistics on correlated properties.

This analysis found an extreme outlier near SeaTac airport. It consisted of a single measurement with four unique BCCH values: 7 for MSTXPWR (all others had values 0-5), 66 for T3212 (all others 9 or 10), 18 for RXACCMIN (all others 0-12), and 1 for CRH (all others 2-4). This base station was recorded both before and after this outlier was measured, and those BCCH properties were different and within the normal range. All these properties are thought to be abused by cell-site simulators. However, some, like the T3212 property, were expected to be lower than normal. This illustrates the strength of our approach of finding anomalies rather than searching for particular values that could be based on false assumptions on how cell-site simulators work. Before having confidence this was an actual cell-site simulator event, and not a abnormally behaving base station, we need additional corroborating evidence or more similar measurements.

5.4 Channel and Location Area Inconsistency

Base stations also advertise location-based properties such as their transmission channels and location area code (LAC). Both of these properties are based on the nearby geography of the BTS and will stand out if they differ from those of surrounding base stations. To avoid detection, a cell-site simulator would need to automatically adjust these parameters, or be manually configured, to conform to its geography. It may also differ from its surroundings on purpose to trigger specific behavior from phones (e.g., broadcasting different LAC to trigger a location update request).

Channel use can also indicate suspicious behavior because networks tend to lease continuous channel blocks that are used by all base stations of that provider. A cell-site simulator may purposely transmit on an unused channel because then it will not interfere with other base stations, including those that it is mimicking.

We did not find any mismatches between the network, location and channel in our data. In Seattle, 100% of the AT&T base stations were on ARFCNs between 128-626 and T-Mobile was measured only on 732-792. Similarly, in Milwaukee 100% of AT&T was transmitted between 180-512 and T-Mobile between 562-786. There were unusual channels found, but further inspection showed the measurements were from BTSs across lake Michigan that have different channel allocations.

We did find anomalies by looking at BCCH broadcasts of the same BTS that were transmitted on multiple ARFCNs. There was a single BTS south of Seattle (Fig. 11) near the United States Citizenship and Immigration Services (USCIS) building — a branch of the Department of Homeland Security — that was recorded transmitting its BCCH on six different channels over two months. This was notable because 96% of all other base stations were found to transmit on a single channel and the other 4% on 2-3 channels.

This multi-channel anomaly is even more strange because 5 of the 6 channels are only found in a 0.4 km area next to the USCIS building, while the bulk of the measurements were made across 5 km, all on the same channel (green in the figure). This pattern would be consistent if a stationary, mimicking cell-site simulator was operating in the USCIS building, but further investiga-

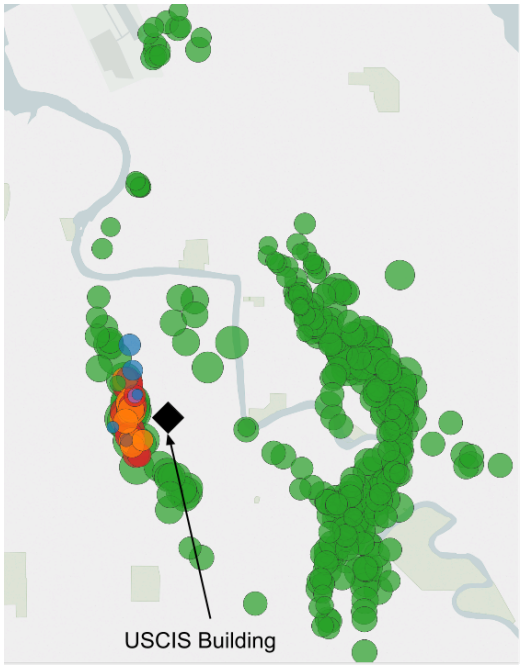

Fig. 11. Map of measurements near United States Citizenship and Immigration Services (DHS) building where the same identifiers transmitted on six different channels in two months. Different ARFCN are indicated by color, and point size indicates signal strength.

tion is needed for a more conclusive characterization of this anomaly.

5.5 Bait Phone Logs

The final signature class that we consider are suspicious events from the perspective of mobile phones. Since the target of a cell-site simulator is a phone, we would expect a cell-site simulator to interact differently with a mobile phone than a normal base station would (e.g., use a weak cipher mode). We take advantage of the SnoopSnitch Android app, created by SRLabs, that logs and categorizes suspicious BTS-to-mobile communications [29]. These mobile signatures are not the focus of this work, but we include them as an independent source of information that can be used to corroborate other findings.

We configured the SnoopSnitch bait phones included with our sensors to function on 2G, 3G, and 4G networks with T-Mobile SIM cards. None of the bait phones detected unusual events within the city limits of Seattle or Milwaukee. There was one anomaly detected far outside of Milwaukee; however since we did not canvass that area, we could not investigate it further.

5.6 Summary

The results of this section show that the detection methods successfully identified anomalies on real data, across multiple dimensions, consistent with the the cell-site simulator signatures we categorized in §2.3. This achieves our goal of deploying a scalable, cost-effective system, using ridesharing drivers, which is capable of detecting the signatures of cell-site simulator across a city. However, without access to ground truth, we cannot definitively classify these anomalies as cell-site simulators. That would require additional corroborating evidence (e.g., detailed public records requests) or suspicious activity seen over longer periods, preferably across multiple independent signatures.

The software used in this study is available at *seaglass.cs.washington.edu*. Researchers interested in analyzing this data for scientific purposes, to study cell-site simulators or other related cell-phone surveillance technologies, should email *seaglass@cs.washington.edu*.

6 Prior Work

There have been attempts to detect evesdropping or tracking by cell-site simulators with smartphone applications, including the SnoopSnitch app developed by SRLabs that we included in the SeaGlass bait phones [1, 29]. These Android apps fingerprint BTS properties and mobile-to-base stations communications to flag base stations that appear suspicious. Phones must be rooted to access low level base band details, so these apps are unlikely to be widely adopted. As of now, there has been no attempt to crowdsource the data collected by these apps to do aggregated, large scale analyses.

Dabrowski et al., prototyped an Android application that used the topology of the cellular network as a feature to detect cell-site simulators, but they do not report any detailed analysis or deployment results [6]. They also placed four stationary sensors on buildings, which included similar Telit modems as the one used in this study, to fingerprint cell-site simulators. They reported a few irregularities in the frequencies of base stations that were recorded but did not believe these were indicative of cell-site simulators.

There have also been attempts to detect cell-site simulators with the help of cellular carriers. FBS-Radar is recent a system included in the Baidu Phone Guard cell phone app (Android and iOS) that collaborates with mobile carriers to detect spam or fraudulent SMSes sent

by unauthorized base stations [18]. They crowdsource to identify the location of these spamming base stations and report the locations to law enforcement for takedown. FBS-Radar is limited to detecting illegitimate base stations that send spam or fraudulent SMS, and cannot detect less conspicuous attacks, like phone identification, localization, or eavesdropping, that are most commonly used by law enforcement. They were successful at identifying low cost, SMS spamming base stations used by criminals, but it is unclear how well these detection methods would generalize to more sophisticated commercial models used by governments.

Dabrowski et al. also proposed changes to the carrier network monitoring infrastructure to recognize attacks on phones [5]. They suggest that carriers monitor for signs of unexpected location updates, monitoring device authentication round trip time, and cipher downgrade attacks. Working with carriers seem promising, especially against cell-site simulators run by criminals, but given the close relationships between governments and carriers outside the U.S. (e.g., they are often government-owned), they may be less motivated to detect or secure phones against government surveillance.

Finally, there has been work to retrofit the insecure cellular authentication protocols to prevent some cell-site simulator attacks, such as capturing IMSIs [32]. However, this approach has not been implemented by any network providers and would require changes to provider network authentication servers and device SIM cards.

7 Conclusions

Little is currently known about cell-site simulator usage. In the U.S., for example, much of the public's knowledge has been obtained through public record requests or courtroom proceedings. We designed SeaGlass to be a cost-effective approach using robust vehicular sensors to canvass large cities. We deployed SeaGlass sensors on 15 ride-sharing vehicles in two cities, collecting two months of data in each city. We further developed techniques to analyze this data, with the ability to surface anomalies with low false-positive rates. Our results show that the system is capable of detecting anomalies across a wide variety of signature classes, potentially caused by actual cell-site simulators. If cell-site simulators are regularly used, then our results suggest that SeaGlass, or a similar system built on crowdsourced data collection and our signature detection methods, would detect them. Further, the cost of our equipment (roughly $500 per SeaGlass sensor), and our plans to make detailed instructions public, can democratize the ability to monitor cell-site simulator usage, thereby empowering everyday citizens to contribute to the global monitoring of inappropriate privacy intrusions posed by cell-site simulators.

Acknowledgements

This research was supported by the John S. and James L. Knight Foundation Prototype Fund and the Short-Dooley Professorship. We kindly thank the University of Washington Tech Policy Lab, the Knight Foundation, Melody Kadenko, and all of our volunteer drivers for making this project possible. We would also like to thank our anonymous reviewers for their helpful feedback and our shepherd Damon McCoy for getting the paper ready for submission. Finally, thanks to Gennie Gebhart and Paul Vines for critiquing draft versions of the paper.

References

[1] Android-imsi-catcher-detector. https://github.com/CellularPrivacy/Android-IMSI-Catcher-Detector. Accessed: 2017-03-14.

[2] Apple location services. https://support.apple.com/en-us/HT203033. Accessed: 2017-03-14.

[3] D. Barrett. Americans' cellphones targeted in secret u.s. spy program. http://www.wsj.com/articles/americans-cellphones-targeted-in-secret-u-s-spy-program-1415917533, November 2014. Accessed: 2017-03-14.

[4] S. Biddle. Long-secret stingray manuals detail how police can spy on phones. *The Intercept*, September 2016.

[5] A. Dabrowski, G. Petzl, and E. R. Weippl. The messenger shoots back: Network operator based imsi catcher detection. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 279–302. Springer, 2016.

[6] 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*, pages 246–255. ACM, 2014.

[7] C. Farivar. Cops must now get a warrant to use stingrays in washington state. https://arstechnica.com/tech-policy/2015/05/cops-must-now-get-a-warrant-to-use-stingrays-in-washington-state/. Accessed: 2017-03-14.

[8] C. Farivar. City cops in disneyland's backyard have had "stingray on steroids" for years. http://arstechnica.com/tech-policy/2016/01/city-cops-in-disneylands-backyard-have-had-stingray-on-steriods-for-years/, January 2016. Accessed: 2017-03-14.

[9] Fcc antenna structure registration. https://www.fcc.gov/help/antenna-structure-registration-asr-overview.

Accessed: 2017-03-14.

[10] Forcing phones to transmit at high power: 10-08-23-2010 fl v. thomas, 2008-cf-3350a, suppression hearing transcript re: Harris stingray and kingfish. https://www.documentcloud.org/documents/1282618-10-08-23-2010-fl-v-thomas-2008-cf-3350a.html. Accessed: 2017-03-14.

[11] R. Gallagher. Meet the machines that steal your phone's data. http://arstechnica.com/tech-policy/2013/09/meet-the-machines-that-steal-your-phones-data/2/, 2013. Accessed: 2017-03-14.

[12] Google location services. https://developers.google.com/maps/documentation/geolocation/. Accessed: 2017-03-14.

[13] B. Heath. Police secretly track cellphones to solve routine crimes. *USA Today*, August 2015.

[14] Iden transceiver operations manual. https://www.documentcloud.org/documents/3105641-iDEN-2-4-Operator-Manual.html#document/p1. Accessed: 2017-03-14.

[15] Intercept surveillance catalogue - direction finding systems. https://theintercept.com/surveillance-catalogue/category/direction-finding-systems/. Accessed: 2017-03-14.

[16] Intercept surveillance catalogue - stingray iii. https://theintercept.com/surveillance-catalogue/stingray-iii/. Accessed: 2017-03-14.

[17] A. E. Kramer. Ukraine's opposition says government stirs violence. *The New York Times*, January 2014.

[18] Z. Li, W. Wang, C. Wilson, J. Chen, C. Qian, T. Jung, L. Zhang, K. Liu, X. Li, and Y. Liu. Fbs-radar: Uncovering fake base stations at scale in the wild. *Proceedings of the 2017 Network and Distributed System Security Symposium (NDSS'17)*, 2017.

[19] Microsoft Geolocation API. https://msdn.microsoft.com/library/windows/apps/br225603. Accessed: 2017-03-14.

[20] P. Mocek. Cell site simulator acquisition and use (tacoma police department). https://www.muckrock.com/foi/tacoma-72/cell-site-simulator-acquisition-and-use-tacoma-police-department-12243/, June 2014. Accessed: 2017-03-14.

[21] Mozilla location services. https://location.services.mozilla.com/. Accessed: 2017-03-14.

[22] New York Civil Liberties Union. NYPD has used stingrays more than 1,000 times since 2008. https://www.nyclu.org/en/press-releases/nypd-has-used-stingrays-more-1000-times-2008, February 2016. Accessed: 2017-03-14.

[23] Opencellid. http://opencellid.org/. Accessed: 2017-03-14.

[24] Open mobile network. http://www.openmobilenetwork.org/. Accessed: 2017-03-14.

[25] Qrc surveyor 500. http://www.qrctech.com/Surveyor500-p/q3850.htm. Accessed: 2017-03-14.

[26] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[27] J. Scahill and M. Williams. Stingrays a secret catalogue of government gear for spying on your cellphone. https://theintercept.com/2015/12/17/a-secret-catalogue-of-government-gear-for-spying-on-your-cellphone/, December 2015. Accessed: 2017-03-14.

[28] Skyhook wireless. http://www.skyhookwireless.com/. Accessed: 2017-03-14.

[29] SR Labs. Snoopsnitch. https://opensource.srlabs.de/projects/snoopsnitch. Accessed: 2017-03-14.

[30] SR Labs. Snoopsnitch - IMSI Catcher Score. https://opensource.srlabs.de/projects/snoopsnitch/wiki/IMSI_Catcher_Score.

[31] Unwired labs. https://unwiredlabs.com/. Accessed: 2017-03-14.

[32] F. van den Broek, R. Verdult, and J. de Ruiter. Defeating imsi catchers. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 340–351, New York, NY, USA, 2015. ACM.

[33] B. Vielmetti. Groups decry milwaukee police's warrantless use of 'stingray' tracking. *Milwaukee Wisconsin Journal Sentinel*, February 2016.

[34] Washington state law: Hb 1440 - 2015-16. http://app.leg.wa.gov/billsummary?BillNumber=1440&Year=2015. Accessed: 2017-03-14.

[35] Wikileaks - 3g interception / gsm inbetween interception. https://wikileaks.org/spyfiles/docs/ability/80_3g-interception-gsm-inbetween-interception-system.html. Accessed: 2017-03-14.

[36] Wisconsin aclu public records request. https://assets.documentcloud.org/documents/2811521/Milwaukee-PD-StingRay-Use-Log-Sep2015.pdf. Accessed: 2017-03-14.

[37] K. Zetter. Turns out police stingray spy tools can indeed record calls. https://www.wired.com/2015/10/stingray-government-spy-tools-can-record-calls-new-documents-confirm/, October 2015. Accessed: 2017-03-14.

A Sensor Parts and Cost

Part	Cost
Telit GT-864 QUAD/PY GSM modem	**$65**
External antenna	**$25**
Raspberry Pi 2B+2	**$35**
GPS (GlobalSat BU-353)	**$30**
Bait Phone (Motorola Moto-G 4G LTE)	**$95**
4G Hotspot (ZTE Z917) + 3 month plan	**$100**
DC/AC inverter	**$26**
Powered USB Hub	**$17**
Pi accessories	**$15**
SD Card (32 GB)	**$17**
Modem accessories	**$30**
Cables	**$35**
Box	**$12**
Total	**$502**

Table 2. Sensor parts and cost breakdown.

B Base Station Modeling Details

We use the log-distance path loss model as the starting point (Equation 1). This models how the strength of a BTS signal decays as a function of distance from the base station. The inputs are the distance from the BTS in meters (d), path decay (γ), and BTS transmit power in dBm (A). The output is the received signal strength in dBm (s).

$$PL_{A,\gamma}(d) \rightarrow s = A + \gamma log_{10}(d) \quad (1)$$

We transform Equation 1 into a probabilistic model of received signal strength to account for noise (Equation 2). We assume that the strength of each recorded measurement, at some distance d from the base station, is drawn from a normal distribution (with mean derived from Equation 1 and a standard deviation of 5dB). The standard deviation falls within the suggested literature range and matched cases where a sensor held the same position and recorded multiple measurements [26]. The path loss constant γ is set to 3.5, a standard for an urban enviornment [26].

Measurements are defined by the tuple $m = (m_x, m_y, m_s)$, which denotes measurement longitude, latitude and received strength, and BTS parameters are defined by $t = (t_x, t_y, t_A)$, denoting the BTS longitude, latitude and transmit power. With this set up, we can now express the likelihood of measurement point m given BTS parameters t as:

$$P(m|\ t) = P(\mathcal{N}(PL_{t_A,3.5}(d_{m,t}), \sigma = 5)) \quad (2)$$

where $d_{m,t}$ is the distance between the estimated BTS position and measurement in meters.

Let M_t denote the set of all measurements of a given BTS t. Then the total likelihood for M_t is the product of (2) across all measurements, and the corresponding log likelihood is:

$$\log(\mathcal{L}(t|M_t)) = \sum \log(P(m_i|\ t)) \quad (3)$$

Using this formulation, we can analytically optimize t_A in terms of M_t and the BTS longitude and latitude t_x, t_y. This is done by differentiating Equation 3 and finding A that sets it equal to 0. An analytical solution for t_x and t_y is difficult to find, so we use a local search instead. The local search starts with t_x and t_y set to the longitude and latitude of the highest strength measurement in M_t. The search iterates, making small changes to t_x and t_y in each round, to determine if they result in a higher log likelihood. When there are no small changes to the estimated BTS position that increase the log likelihood, the search terminates and the current values of t_x and t_y are used as the final estimates of latitude and longitude.

The resulting method performs well when there are enough well distributed measurements for a base station but there are some limitations. The model does not incorporate locations where the BTS was not measured (the threshold for the GSM modem is -108 dBm). Also, in practice, the actual received strength can be more noisy than a log-distance path loss model, especially across water, which violates the urban modeling assumption. There are more complete models (e.g., Hata-Okumura model) that would require additional complexity and more parameters to infer.

C BCCH Property Correlations

BCCH Property	BCCH Value	T-Mobile USA	AT&T Mobility
ALPHA	8	100%	0%
	10	0%	100%
DRXMAX	0	0%	93%
	4	100%	7%
CTRLACK	1	100%	100%
MSTXPWR	0-4	100%	0%
	5	0%	100%
	7	1 anomaly	0%
NCO	0	100%	100%
NOM	2	100%	100%
PAT	6	100%	100%
PBCCH	0	100%	100%
PCMEASCH	0	100%	100%
PENALTYT	0	0.07%	100%
	31	99.93%	0%
RAC	71-92, 130	100%	0%
	101-124, 131-132	0%	100%
RXACCMIN	0	0%	100%
	4-18	100%	0%
SPGC	0	0%	100%
	1	100%	0%
T3212	9	0%	100%
	10	100%	0%
	66	1 anomaly	0%

Table 3. A selection of measured values for BCCH properties showing the high correlation between the network and the value. Note the two anomalous values for T3212 and MSTXPWR, which are from the same anomalous point measured at SeaTac airport.