Privacy Enabled Crowdsourced Transmitter Localization Using Adjusted Measurements

Harsimran Singh*, Shamik Sarkar*, Anuj Dimri*, Aditya Bhaskara*
Neal Patwari*, Sneha Kasera*, Samuel Ramirez†, Kurt Derr†
* University of Utah
Email: hsingh4@cs.utah.edu, shamik.sarkar@utah.edu, anuj@cs.utah.edu, bhaskara@cs.utah.edu
neal.patwari@ece.utah.edu, kasera@cs.utah.edu
† Idaho National Labs
Email: samuel.ramirez@inl.gov, kurt.derr@inl.gov

***Abstract*—We address the problem of location privacy in the context of crowdsourced localization of spectrum offenders where participating receivers report received signal strength (RSS) measurements and their location to a central controller. We present a novel approach, that we call the *adjusted measurement* approach, in which we generate pseudo-locations for participating receivers and report these pseudo-locations along with adjusted RSS measurements as if the measurements were made at the pseudo-locations. The RSS values are adjusted by representing those as a weighted linear combination of the RSS values at the receivers, where receivers closer to the false location have a higher weight than those far away.**

We use two RSS datasets, one from a cluttered office (indoor) and another from roadways in Phoenix, Arizona (outdoor) to evaluate our approach. We compare the localization error of our approach with that of the naive approach that simply adds noise to locations. Our results demonstrate that location privacy can be preserved without a significant increase in the localization error. We also formulate an adversary attack that attempts to solve the inverse problem of determining the true locations of the receivers from their false locations. Our evaluations show that the adversary does no better than random guessing of true locations in the monitored area.

I. Introduction

Many users are willing to have their networked devices participate in distributed sensing and data collection applications. As an example, car drivers would like to report traffic conditions, and car velocities, if that allows city and state authorities to better plan the road network. Such reporting can also lead to short-term gains for car drivers in terms of a centralized service providing safer, less congested routes. Such a distributed and crowdsourced/crowdsensed data collection is expected to grow in the future. However, the participants may be seriously concerned about their privacy. They do not wish to have their data associated with their identities or locations. In many cases, a participating node reports its identity, its location, and its measurement (car velocity measurements, could also be radio frequency measurements, etc.) to a central controller which then makes this data available for different applications. However, such a data collection system does not necessarily preserve the location privacy of the participating users [1], [2], [3]. Users can be linked to their locations, and multiple pieces of such information over a period of time can be linked together to profile users, which leads to unsolicited targeted advertisements or price discrimination [4]. Even worse, a user's habits, personal and private preferences, religious beliefs, and political affiliations, can be inferred from the user's whereabouts. Therefore, users who are willing to participate in the crowdsourcing system for societal good or some incentives will be uncomfortable or in worst case might not even participate if they feel that their privacy is compromised.

The traditional way to preserve location privacy is to add noise to the location with the hope that the measured data would still be useful and would not severely reduce the quality of the service or the accuracy/utility of the application [5]. Location-based services (LBS) where the application response is based on the user's geographic location [6], [1], [7] use such an approach. Some examples of LBS include location aware task reminder (pick up groceries when near a store), advertisements, and emergency services. For these applications, the coarse location of the user is acceptable. Even for applications like building a weather map of a city, which use both location and the measurements, the measurement does not change over a few hundred meters. Thus, adding noise to the location does not significantly reduce the accuracy of the application. However, for some applications the measurements are closely tied to the location, i.e., there might be a significant change in measurements made at two locations only a few meters apart. For example, various localization applications based on wireless sensor networks rely heavily on the accuracy of reported sensor measurement as well as sensor's locations [8], [9], [10]. The traditional method of adding noise as done in LBS may lead to a large drop in utility, when used in applications that are highly sensitive to location information, as the measurements at the false locations are expected to be significantly different from those at the actual locations.

We address the problem of location privacy in the context of crowdsourced localization of spectrum offenders as in the work of Khaledi et al [8]. In this context, participating wireless receivers report their location and the received signal strength (RSS) they measure of signals emitted from transmitters within their range to a central controller. The central controller collects the RSS and location data and feeds this information as input to localization algorithms to localize spectrum offenders.

It is important to note that the central controller has the location data of all the participating receivers.

In this paper, we consider two adversary models where different entities in the system can be adversarial. In our first model, we consider the central controller to be the adversary. In our second model, the central controller is trusted but the third-party applications using the data at the central controller are adversarial and these try to infer the location of the receivers/users from the data. As we show in the paper, the solution to the location privacy problem corresponding to the first adversary model also applies to the second adversary model. Therefore, unless we mention the second model explicitly, in formulating the problem, in developing the location privacy solutions, and in our evaluations, we assume the first adversary model.

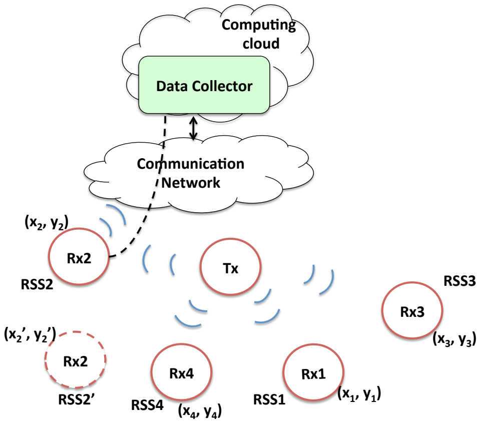


Fig. 1. Adjusted Measurement Idea

We present a novel approach towards solving the problem of preserving a participating receiver's location privacy. In our approach, that we call the *adjusted measurement* approach, we generate pseudo-locations and report the pseudo-locations along with adjusted measurements as if the measurements were made at the pseudo-locations. This idea of adjusting RSS measurements for pseudo-locations is shown in the Figure 1. In this figure, receivers, Rx1-4, are measuring the RSS of the signals transmitted by the transmitter, Tx, and report their locations and the measured RSS values to the data collector. However, for the purpose of protecting its location privacy, an adjusted measurement RSS2$^{'}$ along with the pseudo-location $(x_2^{'},y_2^{'})$ is reported for Rx2 instead of its actual location (x_2,y_2) and its actual measurement RSS2.

The key challenge here is to adjust the RSS measurements of signals from an unknown transmitter suitably to minimize the impact on the transmitter localization accuracy while preserving the participating receiver's privacy. Ideally, the variation of RSS values across the monitored area can be approximated using the path loss model for radio waves [11]. Approximating the RSS values using a path loss model requires knowledge of transmitter's characteristics including transmit power, and antenna gain. However, for localizing an unknown transmitter for the purpose of monitoring, its characteristics are unavailable. Existing work has proposed forming a collaborative group to achieve *k-anonymity* where a reported (location, measurement) pair could belong to any one of the *k* receivers. However, reporting true locations can still lead to privacy violations. Adversaries can correlate the reported locations with other meta information to identify the participating receivers [12]. E.g., a home owner participating in crowdsourcing from his home can be identified by the reported location and directory information. Therefore, it is important that true locations are not reported.

We propose a collaborative approach where the receivers form groups. Within each group, the receivers pick one among themselves as a leader and report their true locations and true RSS measurements to the leader. The leader, then, chooses false locations for these receivers by randomly sampling in a region that includes all the group members and adjusts the corresponding RSS values for the false locations, based on the true RSS measurements it receives. The RSS values are adjusted by representing those as a weighted linear combination of the true RSS values at the receivers within a group, where receivers closer to the false location have a higher weight than those far away. The details of this approach are described in Section IV. Finally, the leader reports the falsified locations and the corresponding adjusted RSS values to the central controller. The leaders of all groups perform the above tasks.

We use two RSS datasets, one from a cluttered office (indoor) and another from the city of Phoenix, Arizona (outdoor) to evaluate our adjusted measurement approach. We also compare the localization error of our adjusted measurement approach with the naive approach which simply adds noise to locations for varying level of noise addition. We observe that the transmitter localization error increases arbitrarily with increasing noise levels for both the datasets for the naive approach. In the indoor environment, we find that the localization error increases from 1.73 meters to 8.5 meters, and in the outdoor environment it increases from 134.24 meters to 232.77 meters. However, our *adjusted measurement* approach significantly reduces this increase in localization error in both the indoor and outdoor settings. Specifically, in the indoor environment, with location noise uniformly distributed in (-14, 14) meters along both latitude and longitude, the localization error reduces from 8.5 meters to 3 meters. In the outdoor environment, with location noise uniformly distributed in (-350, 350) meters along both latitude and longitude, the localization error reduces from 232.77 meters to 167.02 meters. Our method using randomly selected location for receiver has an error of 1.8 meters and 155.60 meters in indoor and outdoor setting respectively. We also formulate an adversary attack that attempts to solve the inverse problem of determining the true locations of the receivers from their false locations. Our evaluations show that the adversary does no better than random guessing of true locations in the monitored area.

II. Adversary Model

We consider the following two adversary models in this paper.

A. Model 1: Malicious Central Controller

In our work, users (corresponding to the receivers) wish to protect their location information from the central controller. Thus, we treat the controller as an adversary. The controller has access to all the readings of the receivers participating in the crowdsourcing system. These readings are reported in the form *(Timestamp, Location, RSS),* where Timestamp is the time when a measurement is made, and Location is an (x, y) tuple representing the latitude and longitude of the receiver's location. We also assume that the central controller knows the number of participating receivers and the algorithm used to adjust the RSS measurements for the false location.

That the adversary has a complete knowledge of the algorithm used to adjust the RSS measurements is an important assumption. Since the output of this algorithm is a function of the true receiver locations and the true RSS values, the adversary could try to reverse engineer the algorithm and find out the true locations of the receivers (indeed, we consider such an attack in Section VI).

We assume that the adversary does not deploy nodes to locate the participating receivers when they are sending measurements to their leader or the central controller as such a threat could exist with or without adjusted measurement approach. In our proposed collaborative approach, we assume that the communication in each receiver group is secure, and that the participants, including the group leader, are trustworthy, and that they do not collude with the adversary (i.e., the central controller).

B. Model 2: Malicious Third Party Applications

In this adversary model, the central controller is not an adversary but the applications that use the collected data are adversarial. This adversary model then would allow the participating receivers to report their true data and location to the central controller which now adjusts the measurements. Such a model does not require group members to trust any leader receiver or each other. This model represents many scenarios in which users are willing to trust a central service but not necessarily other peers. Moreover, the receivers in this model need not communicate with each other. However, very importantly, our methodology for adjustment of measurements, that we develop for the first model, applies to this second model as well.

Note that each of the two models is considered disjointly and not in conjunction with the other model.

III. Privacy Definitions

To measure the location privacy of a user, we use the following two metrics:

1) *k-anonymity:* This is one of the most widely used privacy metrics. Simply put *k-anonymity* means that an adversary can narrow the identity of an individual down to a set of k people, but no smaller [13]. In our proposed solution, the central controller will be able to associate a measurement with a group, but not to any smaller subset of receivers in the group. So, our method achieves k-anonymity, where k is the number of receivers in the group. By increasing k, we reduce the adversary's ability to associate a measurement to a single receiver or user.
2) *Proximity to true locations:* Another way to measure privacy is by the extent to which the adversary can determine/guess the receiver locations. In our evaluation, we consider the "matching cost" between the true locations and the adversary's guesses. For formal definitions, we refer to Section V.

IV. Methodology

In this section, we first describe a simple noise addition approach followed by our proposed adjusted measurement approach and then our final method of random selection of false locations with adjusted measurements. In the simple noise addition approach, receivers simply add noise to their true locations in order to protect their privacy.

A. Naive Approach: Adding Noise

A simple way for a user to preserve his/her location privacy is to report a false location. Specifically, for some chosen noise level, the user can report a latitude, lat, and a longitude, lon, given by the following equations

$$\textit{lat} = \textit{lat} + \textit{random.uniform(-noise\_level, noise\_level)} \quad (1)$$

$$\textit{lon} = \textit{lon} + \textit{random.uniform(-noise\_level, noise\_level)} \quad (2)$$

The higher the noise level, the further the false location is from the true one, on average, and thus higher the privacy. However, as we move away from the true location, the RSS measurements which were made at the true location slowly stop making sense at the false location. Therefore, with increasing noise levels the localization error increases rapidly (see Figure 5(a)). Besides the drop in the utility (transmitter localization), this approach has other concerns as well. Since the location is reported by the device itself and the false location is obtained by adding random noise to the true location, there is a possibility of averaging and linkage attacks by the central controller [12]. Furthermore, in this approach, the user is unaware of the number of other users in the system in its vicinity. Thus, the user may face difficulty in choosing the right noise level for the desired privacy guarantee.

B. Adjusted Measurements

It is challenging to preserve utility (i.e. localization accuracy) while reporting false locations. Our approach towards achieving both utility and privacy is to report a noisy location, while carefully adjusting the reported RSS measurement. While natural, this idea can be tricky to implement. The RSS field in a region can exhibit a complex behavior, and thereby, making it difficult to determine a *plausible* RSS value at the false location. For instance, Fig. 4(a) and Fig. 4(b) show the contour plot for the RSS field in a cluttered office space area.

The transmitter locations in these figures are (5.5, 4.1) and (3.2, 9.1), respectively. Note that the RSS contour lines deviate significantly from the concentric circles expected from a standard log-path-loss model thus showing the contours in the real world can be complex. This makes modeling the variation of RSS values across the monitored area appropriately with the help of well-known propagation models harder/in-feasible. Moreover, given that we do not have any prior information on the offending transmitters characteristics including transmit power, antenna type, angle etc., using path loss model which rely on transmitter location/characteristics is not possible.

Our proposal to overcome this challenge is to use *collaboration* among small groups of receivers (i.e., users). Our idea is illustrated in Fig. 2. The participating receivers form a group and select a leader. They then report their (loc, rss) pair to the leader, who is responsible for reporting these readings to the central controller. After receiving the readings from all the users in the group, the leader then chooses a false location for each receiver by adding some noise (similar to the simple noise addition method) but also adjusts the RSS measurements for the false location. The leader then reports these false locations and their corresponding adjusted RSS to the central controller.

At first, it seems that we have simply transferred the difficulty of estimating RSS values at the false locations to the leaders. However, the key advantage now is that a leader has access to the RSS measurements at k different locations in a region, and can thus *interpolate* (described below) to estimate the RSS values. Also, a leader reporting values has other advantages: the adversary has no way to determine which user a particular measurement belongs to; this inherently prevents averaging and linkage attacks.

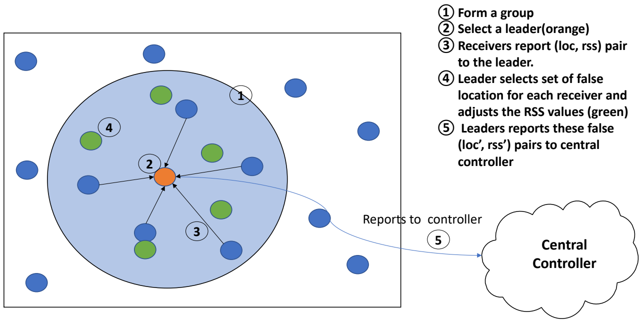

① Form a group
② Select a leader(orange)
③ Receivers report (loc, rss) pair to the leader.
④ Leader selects set of false location for each receiver and adjusts the RSS values (green)
⑤ Leaders reports these false (loc', rss') pairs to central controller
1
2
3
4
Reports to controller
5
Central Controller

Fig. 2. Adjusted Measurement Approach

We now describe the interpolation procedure that our leader uses. Our method is based on a simple yet powerful idea: the RSS value at a desired location is closer to the RSS values at locations *near* it than those far away. Therefore, given a false location f (at which the RSS value is unknown), we may express the RSS value at f as a weighted linear combination of the RSS values at receivers, where the receivers closer to f have a higher weight than those far away. Specifically, if we have a receiver at a distance d from a point f as above, the RSS value of the receiver has a contribution proportional to w_i to the estimated RSS value at f. Formally, if we have n receivers in an area, the RSS value at a false location f is:

$$RSS_f = \frac{\sum_{i=1}^{n} w_i RSS_i}{\sum_{i=1}^{n} w_i}. \tag{3}$$

We tried two weighing methods. In the first method, $w_i = {d_i}^{-c}$ where d_i is the distance between the i^{th} receiver and false location f and c is a constant dependent on the environment. This constant determines how quickly the signal strength decays. It tends to be higher in a obstructed area, such as a downtown area, than a relatively open environment, such as a flat rural area. In the second method, $w_i = e^{-d_i/c}$, where d_i is the distance between the i^{th} receiver and false location f and c is a constant equal to half the average distance between neighboring receivers. Figure 3 shows the error in estimated RSS with varying group size for both interpolation methods. Since the ${d_i}^{-c}$ method is slightly better, we use it as our method for interpolation for the remainder of this paper.

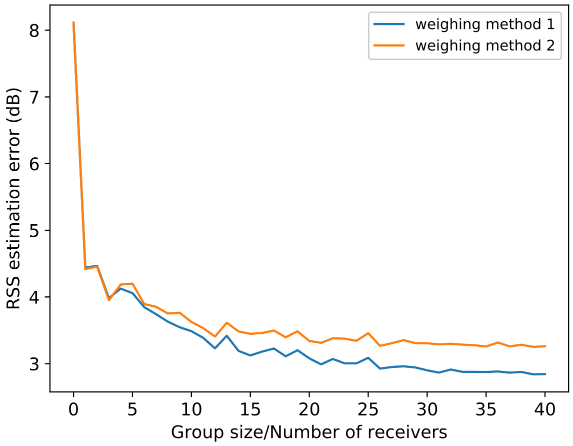
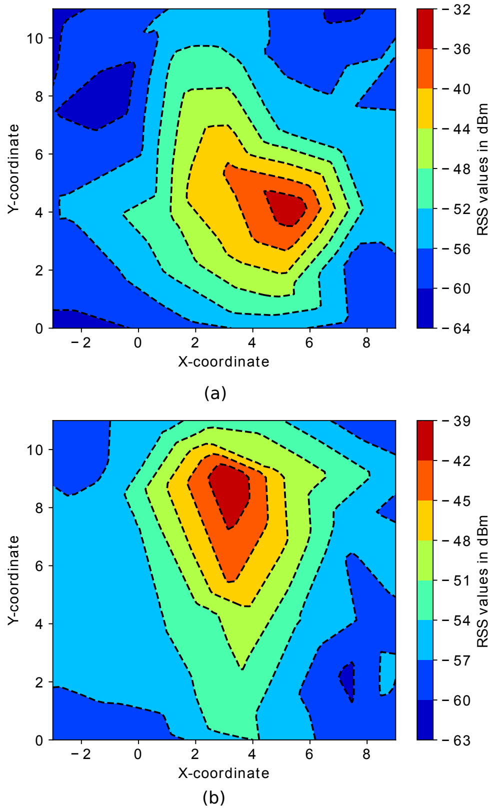


Fig. 3. Error in RSS value estimation for different methods of interpolation (a) weighing method 1: $w_i = d_i^{-c}$ (b) weighing method 2: $w_i = e^{-d_i/c}$

The method to choose false locations for the receivers and adjustment of RSS values is summarized in Algorithm 1. In Algorithm 1, the variable $totatWeights$ is the denominator of eq. 3 and the variable $WeightedRSS$ is the numerator. These false locations and adjusted RSS values are then reported to the central controller.

C. Our Method: Adjusted Measurement with Random Locations

The final algorithm, we propose is a slight variant of the one above. The users form groups, and each group elects a leader, who report the perturbed locations of the points, along with the RSS measurements computed as above. However, to choose a perturbed location of a receiver, we do not add noise to its true location, but instead take a more global approach.



(a)

(b)

Fig. 4. RSS field contour plots in an area

Our approach also hides the precise number of users in the group.

Formally, we consider the spatial region R corresponding to the group (in our experiments, we use a slightly enlarged bounding box), and we select $k \leq n$ *random locations* from R as the points we report. We then use the interpolation procedure described above to compute the RSS values to report. The details can be found in Algorithm 2. This approach certainly preserves privacy better (as we now do not give out information such as the approximate positions of the receivers or even their number). As we see in our experiments, it does not reduce the utility in any significant manner.

V. Evaluation and Results

For the evaluation of utility, our baseline (the 'gold standard') is the localization accuracy when the true location and RSS measurements are reported by the receivers to the central controller but all privacy is lost. We compare our adjusted measurements approach against the simple noise addition approach, in both indoor and outdoor settings.

A. *Indoor: Cluttered Office Space*

For our indoor experiments, we use public data [14] that was collected in an office area that is cluttered with desks, bookcases, filing cabinets, computers, and equipment. In the experimental set up for this data collection, 44 sensors were placed randomly in a 15m by 14m area. The sensors transmitted sequentially. When one sensor transmitted, an RSS measurement was made by all the remaining sensors. Each of the 44 sensors transmitted once thereby providing us 44 transmitter locations.

Algorithm 1 AdjustedMeasurement

```
procedure AdjustedMeasurement($receiver\_list, noise\_level$)
    newLoc ← []
    newRss ← []
    for receiver in receiver_list do
        latitude ← receiver.latitude
        longitude ← receiver.longitude
        latitude ← latitude + random.uniform(-noise_level, noise_level)
        longitude ← longitude + random.uniform(-noise_level, noise_level)
        weightedRSS ← 0
        totalWeights ← 0
        for recv in receiver_list do
            distance ← euclideanDist(latitude, longitude, recv.latitude,
                        recv.longitude)
            if distance == 0 then
                weightedRSS ← recv.rssVal
                totalWeights = 1
                break
            weightedRSS ← weightedRSS + $1/d^c$ * recv.rssVal
            totalWeights ← totalWeights + $1/d^c$
        modifiedRSS ← weightedRSS / totalWeights
        newLoc.append((latitude, longitude))
        newRss.append(modifiedRSS)
    return newLoc, newRss
```

Algorithm 2 Adjusted Measurement with Random Sampling

```
procedure RandomSample(R, receiver_list, num_to_sample)
    Randloc ← []
    AdjRSS ← []
    for (i=0; i < num_to_sample; i++) do
        latitude ← random.sample(R.xmin, R.xmax)
        longitude ← random.sample(R.ymin, R.ymax)
        weightedRSS ← 0
        totalWeights ← 0
        for recv in receiver_list do
            distance ← euclideanDist(latitude, longitude, recv.latitude,
                        recv.longitude)
            if distance == 0 then
                weightedRSS ← recv.rssVal
                totalWeights ← 1
                break
            weightedRSS ← weightedRSS + $1/d^c * $recv.rssVal
            totalWeights ← totalWeights + $1/d^c$
        modifiedRSS ← weightedRSS/totalWeights
        RandLoc.append((latitude, longitude))
        AdjRSS.append(modifiedRSS)
    return RandLoc, AdjRSS
```

Our aim here is to evaluate the adjusted measurement approach. For every experiment, we consider all the 44 transmitter locations for localization of transmitter and take the average of the localization error of all the 44 locations for each group size. The group size is the number of receivers collaborating with the leader along with the leader himself. We vary the group size to show that with increasing size the localization error generally decreases; this demonstrates the power and potential of the collaborative approach to obtain both privacy and accuracy.

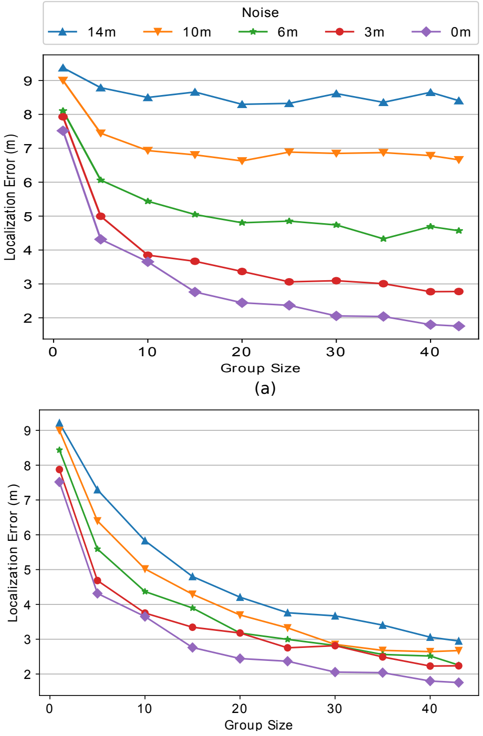


Fig. 5. Localization error for varying group size for different noise levels in simple noise addition and adjusted measurement approach

1) Basic Approach: Adding Noise: Fig. 5(a) shows the results when each receiver simply adds noise to its location before reporting the measured RSS value. The noise is added to the location to get a false location according to (1) and (2). Each user independently adds noise to the location before reporting to the server. We add noise of varying levels in the range (0m, 14m). We can see that the localization error initially drops a bit as the number of participating receivers increase but gradually flatten out at high error values as expected. For large noise levels, the improvement in localization accuracy expected by increasing the number of receivers is almost negligible. With smaller noise the pattern is comparable to noise free localization where with increasing receivers the localization error decreases. However, the final error is still considerably higher than noise-free localization; even with noise_level being 6m, the localization error increases by roughly 2.5 times. This is because the RSS measurements made at the true location stop being meaningful at the false location as we gradually increase the noise levels.

2) Adjusted Measurement: Fig. 5(b) shows the same experiment but with our adjusted measurement approach for varying noise level. We see that the error though initially high with smaller group sizes but as expected gradually decreases as we increase the number of receivers. As is clearly evident this method significantly improves over simple noise addition method. Even in the case of highest noise of 14m, the localization error is under 3m. Also with increasing noise levels, we see a slight increase in the localization error. This is due to the difficulty in predicting the RSS measurements even after using our interpolation method.

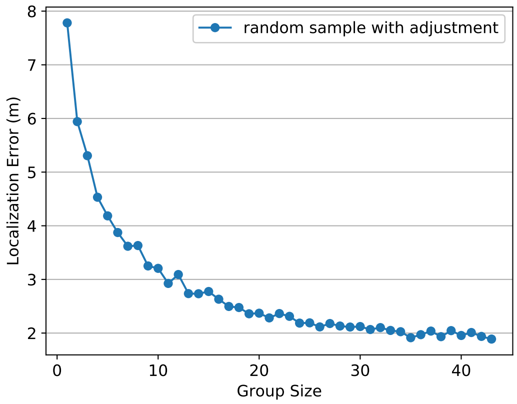


Fig. 6. Random Sampling of False Locations

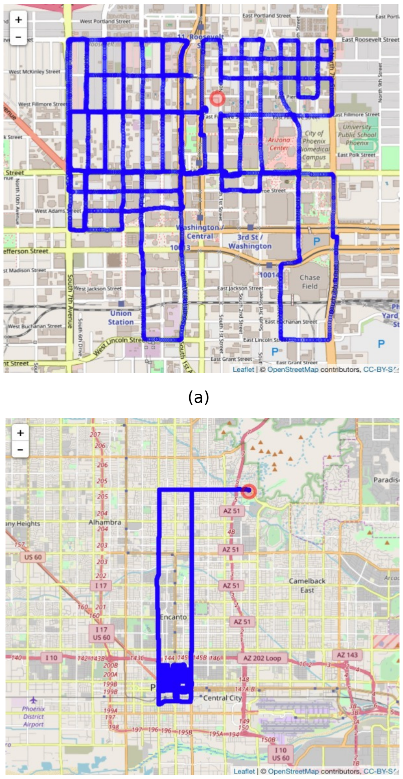
The most interesting case is the one where the false locations are randomly sampled in an area (Algorithm 2) instead of adding noise to the receiver location. In this case, all the false locations are within the area, and therefore, the RSS adjustments are more accurate. Fig. 6 shows the localization error vs. the group size. We can see that the localization error decreases with increasing group size. It also improves the localization error 3m (in case of adjusted measurement approach with 14m of noise level) to 2m. Also with about 25-30 points, the localization error starts to flatten out. Hence, while reporting to the central controller, the leader can sample fewer false locations than actual receivers to minimize information exposed about the true receiver locations to the central controller without comprising the localization accuracy.

B. Outdoor: Phoenix, Arizona

For the outdoor setting, we obtained a dataset collected in the roadways of the city of Phoenix, Arizona. This data is collected by placing a transmitter at a fixed location and driving around the city area and recording the RSS measurements. This is repeated for multiple transmitter locations and the readings are recorded for 1 second at each location. Fig. 7(a) and Fig. 7(b) show two transmitter locations. The red marker shows the transmitter and the blue markers show the location where RSS measurements were recorded for each case.

The outdoor dataset helps us further validate our adjusted measurement approach and shows that it scales well to larger areas without any significant reduction in efficiency or accuracy. In our collaborative set up, we restrict the communication range between the participating receivers in the outdoor, city-scale setting. For our outdoor data set we restrict this range to 350m. Essentially, we select a receiver with a local RSS maxima as the leader in an area, and then select the receivers around it within a range of 350m (see Section VII).

However, if our central controller is non-adversarial and if privacy must be preserved from applications that use the data



(a)

(b)

Fig. 7. Transmitter location (Red) along with receiver positions(Blue)

collected at the central controller, our participating nodes do not need to communicate with each other. Then the adjustment of the RSS values can be shifted from the leader to the central controller.

For the outdoor setting, we compare our adjusted measurement approach with the simple noise addition approach. The performance of the localization algorithm with no noise serves as the performance (localization accuracy) baseline in this comparison. In general, as we move away from the transmitter the RSS values become more noisy. For this reason, considering receivers around the local maxima is a reasonable choice for the purpose of localizing the transmitter. However, from our experiments we observe that, in certain scenarios using only a set of receivers around the local maxima can lead to poorer results (localization accuracy of the transmitter) compared to the case where all the receivers in the monitored area are used. This kind of scenarios occur when the transmitter is at an edge or the number of receivers near the transmitter is small. Once such example scenario is shown in Fig. 7(b). This is justified by the fact that, in scenarios like Fig. 7(b) if the number of receivers around the local maxima is low, contributions from other receivers (that are away from the local maxima) helps in localizing the transmitter. So, the performance of our baseline, i.e. localization algorithm with no noise, is evaluated in the following way. We run our localization algorithm using two methods: (a) using all the receivers in the monitored area and (b) selecting a local maxima and using receivers around it in a fixed radius. Among these two methods, the performance of the superior is considered the baseline. The column labelled “No Noise” in Table1 and Table2 contains the minimum localization error obtained from these two methods.

1) Simple Noise Addition: Table I contains the results for addition of noise with varying noise level. We show results for 250m, 300m and 350m of noise levels. Recall that the noise is added to both latitude and longitude of the receiver where the based on the noise level, the noise is randomly picked from a uniformly distribution in the range(-noise_level, noise_level). As expected with increasing noise levels the localization error increases arbitrarily. In quite a a few cases like case 1 and 6, the localization error almost double with the noise level of 350m. On an average with 350m of noise, the localization error increases by 98.52 meters (approx 76%).

2) Adjusted Measurement approach: The last two columns of Table II contain the results for adjusted measurement approach. We present the result for the highest noise setting i.e., 350m. The results are shown for both cases where false locations are obtained by adding 350m of noise and random sampling. The adjusted measurement approach significantly improves the localization error over simple noise addition approach. We see that on average the localization error drops from 98.52 meters to 32.77 meters for 350m noise addition along with adjusted measurement and 21.36 meters in case of randomly sampled false location with adjusted measurement. It is interesting to see that in certain cases, 1, 2 and 5, the localization error increase is almost negligible. For certain transmitter locations like 7(b) where the transmitter is on the edge, the localization error is relatively high but still significantly less than the simple noise addition method. This is because there aren’t enough receivers around the transmitter to accurately interpolate the RSS values. Also, such cases are unlikely to occur in crowdsourced environment where receivers are spread around more uniformly. Interestingly, the random sample approach is able to bring down errors in certain cases like TX Loc 4 which happens to be a situations like Fig. 7(b).

VI. Adversary Attack

The adversary (in our case, the central controller) receives k readings from a region R by the group leader. The false locations of these readings (as explained in the methodology) are chosen at random from R and therefore, given the locations alone, the only knowledge the adversary could gain is that the group leader is in the region R, and that there are k users in the region. However, note that the adversary is not just aware of the false locations. It also receives the adjusted RSS measurements (which are computed using the *true* locations by the group leader). Thus we ask: can the adversary use the RSS values to set up an inverse problem to solve for the true locations?

A. Inverse Attack

We now consider an attack based on the idea above. Note that for each group, the adversary receives the false locations

TX loc	No noise	Noise 250m	Noise 300m	Noise 350m	Error increase for Noise 350m w.r.t. No Noise
1	136.27	223.28	239.58	270.34	134.07
2	146.58	229.68	234.48	235.11	88.53
3	147.03	229.81	236.22	238.90	91.87
4	101.57	148.21	169.91	193.79	92.22
5	161.41	192.32	208.96	225.18	63.77
6	112.63	219.33	226.57	233.31	120.68
Avg.	**134.24**	**207.10**	**219.28**	**232.77**	**98.52**

TABLE I
LOCALIZATION ERROR (IN M) FOR VARIOUS TRANSMITTER LOCATIONS FOR SIMPLE NOISE ADDITION METHOD

TX Loc	No Noise	Adjusted measurement and Noise of 350m	Error increase for adjusted measurement approach w.r.t. No Noise	Random Sample	Error increase for Random Sample with Adjusted Measurement approach w.r.t. No Noise
1	136.27	147.72	11.45	142.93	6.66
2	146.58	159.76	13.18	147.28	0.70
3	147.03	194.03	47.0	181.17	34.14
4	101.57	170.84	69.27	136.31	34.74
5	161.41	171.08	9.67	168.80	7.39
6	112.63	158.69	46.06	157.14	44.51
Avg.	**134.24**	**167.02**	**32.77**	**155.60**	**21.36**

TABLE II
LOCALIZATION ERROR(IN M) FOR VARIOUS TRANSMITTER LOCATION FOR ADJUSTED MEASUREMENT APPROACH

and the adjusted RSS measurements. The adversary knows the interpolation procedure used to generate the RSS values from the true measurements.

The adversary thus needs to solve the following inverse problem: given the false locations and the corresponding RSS values, what are the different 'configurations' of true locations and RSS values that could have produced them? The adversary's hope is that there are only a few configurations that can *explain* the reported values. But on the other hand, if there several distinct configurations, the adversary has no real way to know which configuration corresponds to the true locations. In our experiments, we show that the latter situation dominates, indicating privacy preservation.

Next, we describe the attack formulation and evaluation in detail. As the different groups have no interaction, we will focus on one group of receivers.

1) Inverse Problem: The controller has access to the following information:

- number of receivers/reported noisy locations (n),
- reported noisy locations (represented by vector L^f),
- corresponding adjusted RSS values (represented by vector R^f), and
- knowledge of the Algorithm 2 (represented by f) used to generate the false locations and the corresponding adjusted RSS values.

Next, we describe the adversary's attack formulation. The symbols used are mentioned in Table III.

Symbol	Description
L^t	vector of true locations of the receivers
R^t	RSS values observed by receivers at true locations L^t
L^f	vector of false locations reported to the central controller
R^f	vector of RSS values reported along false locations L^f
L^a	vector of locations representing adversary's guess of true locations
R^a	vector of RSS values representing adversary's guess of RSS values at L^a
R^c	vector of RSS values calculated at L^f using L^a and R^a
f	Algorithm used to adjust RSS measurements at the false locations

TABLE III
SYMBOLS AND THEIR MEANING

1) Adversary initializes its guess of true locations (L^a) with random guess within the area being monitored. Each element of vector L^a is a tuple comprising of latitude and longitude (lat, long). Based on the reported false locations L^f, the adversary computes the bounding box represented by x_min, y_min, x_max and y_max. The random initialization of each location is done using a uniform distribution as follows:

$$\text{lat = uniform(x\_min, x\_max)} \tag{4}$$

$$\text{long = uniform(y\_min, y\_max)} \tag{5}$$

2) Adversary then initializes the RSS (represented by vector R^a) at locations guesses (L^a) with the RSS value of the the false location which is nearest to the location guess. This is a better initialization than random because RSS values are likely to be similar to locations near to the current location guess rather than some random RSS value.
3) Next using the expression from Algorithm 2, the adversary calculates the RSS values at each of the the reported false locations based on his current guess of true locations and their corresponding RSS.
4) The loss function is defined as the sum of square of the difference between the adversary's calculation of RSS and the actual reported false RSS for each false location reported. The loss (L) is given in eq. 6

$$L = \sum_{i=1}^{n} (R_i^c - R_i^f)^2 \tag{6}$$

The adversary calculates the RSS at the false location using the same method in which the receivers adjust their RSS values before reporting to the central controller.
5) In the next step, to update his i^{th} guess in vector L^a and R^a, the adversary takes the gradient of the L w.r.t to L_i^a and R_i^a. The updates are made according to eq. 7 and eq. 8:

$$L_i^a = L_i^a - \eta \frac{\partial L}{\partial L_i^a} \tag{7}$$

$$R_i^a = R_i^a - \eta \frac{\partial L}{\partial R_i^a} \tag{8}$$

6) This process is repeated again from step 3 till the loss function reduces and flattens out.

The formal algorithm for the attack is given in Algorithm 3, and the results are presented in Section VI-A2.

We observe that one way to achieve zero loss is to set the guesses to be precisely the false location and the corresponding RSS (and of course, this solution does not give any insight into the true locations). The adversary can thus re-initialize the guessed locations that are too close to false locations, and hope that this helps identify the true location. However, we did not observe any improvement in the solution, and thus we stick to Algorithm 3.

2) Evaluation: Figures 8 and 9 show the results from one scenario of transmitter localization. Fig. 8 shows the loss function for one run of the attack. The x-axis shows the iteration number and y-axis the value of loss function at each iteration of the adversary attack. We see that the loss drops to a very low value, implying that the adversary's final guesses explain the RSS measurements at the false location well. Having obtained low loss values, we evaluate how close the adversary's estimates are to the true locations.

Figure 9 shows the final guesses of adversary's attack along with the reported false locations and the true location of the receivers. We can see that even after very low loss, the true locations of the receivers are significantly different from adversary's guess of their true location. Though some of the guessed locations seem quite close to the true locations, the adversary has no way to determine which ones of these satisfy this property.

Algorithm 3 Adversary_Attack

```
procedure Adversary_Attack($false\_location$, $false\_rss$, $num\_recv$, $num\_iters$)
    true_loc = []
    true_rss = []
    for i in range(num_recv) do
        l = rand.uniform(x_min, x_max), rand.uniform(y_min, y_max)
        true_loc.append(l)
    for i in range(num_recv) do
        r = pick the closest false location to the true location values
        true_rss.append(false_rss[r])
    grad_loc = []
    grad_rss = []
    $\eta$ = 0.01
    while num_iters $\geq$ 0 do
        for i in range(num_recv) do
            grad_loc.append($\partial L/\partial true\_loc[i]$)
            grad_rss.append($\partial L/\partial true\_rss[i]$)
        for i in range(num_recv) do
            true_loc[i] = true_loc[i] - $\eta$ * grad_loc[i]
            true_rss[i] = true_rss[i] - $\eta$ * grad_rss[i]
        num_iters = num_iters - 1
    return true_loc
```

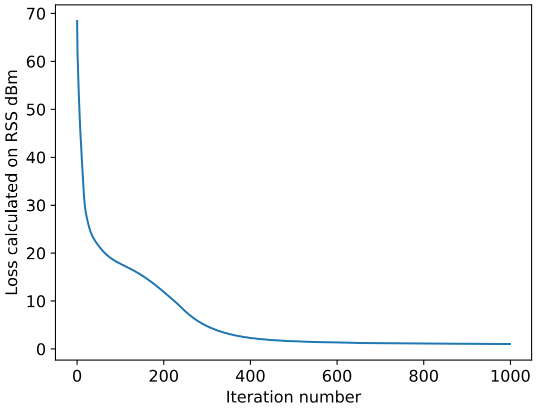


Fig. 8. Loss function vs. iterations of adversary's inverse attack algorithm.

To quantify the proximity of the adversary's guesses to the true locations, we formulate a minimum distance bipartite matching problem. The matching cost is then scaled by the number of receivers in an area: error = minimum matching cost/number of receivers.

Fig. 10 shows such a matching of the adversary's guess to the true location of the receivers.

To determine if the attack above learns some *structure* about the true locations, we compare the matching cost above with the corresponding cost when the guesses are completely random points (uniform and independently) chosen from the given area for 100 runs of adversary attack. 11(a) and 11(b) show the matching cost for 100 runs with random locations and the cost for the final guesses from the attack above, respectively. We can see that the range of the *matching cost* is identical in both the cases. The average *matching cost* for

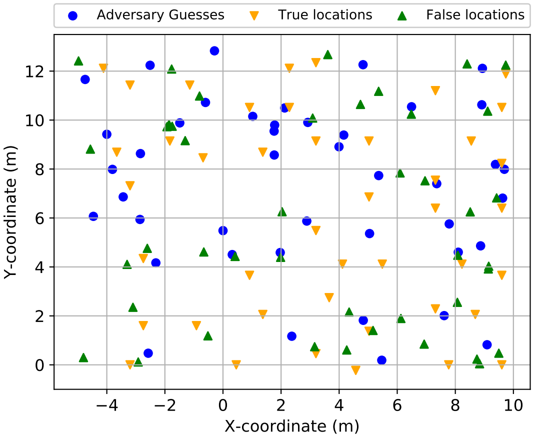


Fig. 9. Adversary guesses, true locations and reported false location in an area.

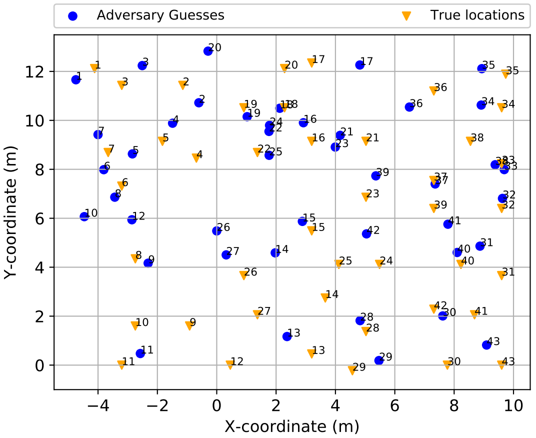
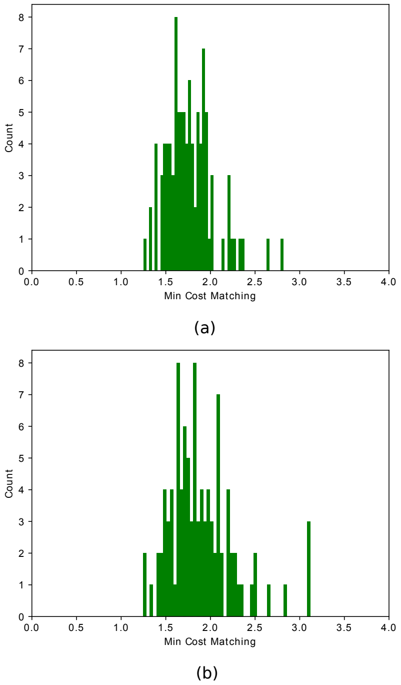


Fig. 10. Mapping of the adversary's guesses to true receiver locations

randomly sampled locations is 1.81 and that of the attack is 1.84. Moreover, we observe that for different runs of inverse attack, each time the final error/loss (equation 6) is a very low value (< 0.1 dB). Thus, the adversary's attack has high uncertainty as the adversary cannot consider a particular pattern to be more likely as the true location of receivers.

We also compute the matching cost between the adversary's guesses and the false locations reported to the controller. Figure 12(a) shows the *matching cost* for 100 runs, as before. We see that the adversary's guesses are not converging to the false locations. This indicates that there are multiple different patterns of locations that are solutions to the inverse problem set up by the adversary (described at the start of the section). This gives further evidence of the privacy preserving nature of our method. Fig. 12(b) shows *matching cost* of the adversary's guesses to one another for various run of inverse attack. Adversary's guess of one run is compared with the guesses from all other runs by taking the *matching cost* between them. We can see that on average the adversary's guesses for each run converge to a different set of location



(a)

(b)

Fig. 11. Matching cost between the true locations and (a) guesses from the adversary's inverse attack and (b) random guesses of true locations

and RSS values. This further strengthens our claim that there exist multiple patterns of location and RSS values which yield low error calculations for false location.

To summarize via the terminology of [15], the adversary attack has a high uncertainty (as there are many different solutions to the inverse problem). It also has a low correctness, as the solutions obtained have nearly the same matching cost as random points.

VII. Practical Considerations

In this section, we consider the practical issues tied to communication between participating receivers for choosing a leader among themselves as well as for communicating with the leader, once the leader is chosen. In indoor environments, we expect users to communicate using WiFi, WiFi Direct, or any short range wireless technology. For outdoor, city-scale settings, we expect users to communicate using a protocol like Dedicated Short Range Communication (DSRC) for vehicular communication. It operates in the 5.9GHz band with an ideal range of 1000 meters [16]. For our evaluation, we choose a DSRC communication range of 350 meters.

If our central controller is non-adversarial and if privacy must be preserved from applications that use the data collected at the central controller, our participating nodes do not need to communicate with each other. They can directly (using cellular or WiFi networks) send their location and measured data to the central controller, which runs the adjustment algorithm, instead of distributed leaders. Thus, under this adversary model, our approach can be implemented without requiring any direct device-to-device communication method.

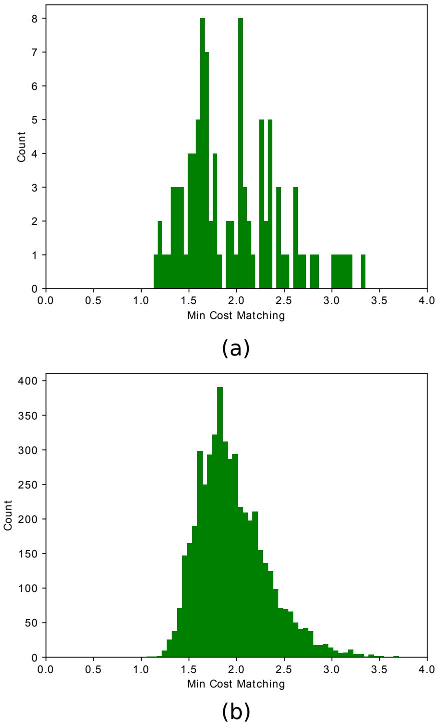


Fig. 12. Matching cost between (a) adversary guesses of true locations and false locations reported (b) adversary guesses to one another for 100 runs of adversary's inverse attack

VIII. Related Work

We present related work in two categories: (a) different threats associated with location sharing and techniques used to preserve the location privacy of the user, and (b) various approaches and applications for transmitter localization.

A. Location Privacy

Sharing of location can have various threats associated with it. These threats are well studied in the existing literature [4], [1], [2], [3], [17]. Users can be identified even if they share their location sporadically [3]. To reduce the threat to location privacy certain applications anonymize or obfuscate their data [18], [19], [20]. However, a knowledge of the social graph of the user (relations among the users) can help an adversary to de-anonymize their location traces [21]. Also, seemingly non-intuitive, location sharing of a user also has the potential to diminish the privacy of others [22].

In the obfuscation approaches, a user can report true identities but instead of the true location, it reports a nearby but false location [19]. Apart from being ineffective in preventing absence disclosure [23], obfuscation based approaches can cause considerable degradation in utility which can be a deterrent in their deployability. k-anonymity approaches have been used to make user indistinguishable from k-1 other users. These also incorporate a user defined privacy level based on the choice of k. Gedik et. al [20] show one such customizable k-anonymity system and alongside implement a spatial-cloaking algorithm which anonymizes the location and cloaks it in an area before forwarding the location information to an LBS server. Collaborative approaches have also been used to preserve privacy for LBS [24], [18]. Shokri et. al [24] describe a collaborative privacy preserving approach called MobiCrowd which forms a peer-to-peer network and only queries the LBS if none of the peers have the required information for a given location. Chow et. al [18] have a similar approach of forming a peer-to-peer network to form a spatial cloaking region. The user can then filter out the results based on its precise location. The above approaches are designed for LBS (where user is the beneficiary of the data/information) and hence, cannot be used directly for 'reporting data' at a false location. For our application, we need to report measurements at a false location, so we use a collaborative approach to adjust the measurements for the false locations before reporting them.

B. Transmitter Localization

Localization of an RF source has been extensively studied over several decades [25], primarily using time and time-difference measurements. RSS measurements preserve privacy in the sense that the receiver does not need to record the received signal itself, which may contain private data, and can provide accurate localization due to the high density of transceivers in our environments, for example using WiFi fingerprinting [26], in sensor networks [14], or as a complement to GPS for cellular localization [27].

More recently, opportunistic spectrum reuse emerged as a means to improve the efficiency of our use of the radio spectrum [28]. A collaborative sensing algorithm can identify the "holes" where secondary use of the spectrum may occur [29]. While [29] simply locates these holes, an alternative approach is to locate the transmitters and identify their gain patterns so that their coverage area can be calculated [30].

Primarily, it is assumed that one transmitter is located at a time, that if multiple transmitters are to be located, their signals can be separated at the receivers. During jamming attacks, a sophisticated adversary can make this impossible. When multiple signals cannot be separated, methods [31], [8] can localize multiple transmitters from RSS measurements. While [31] has relatively high time complexity, and assumes that the number of transmitters is known, [8] estimates the number of transmitters and separates the problem over space into individual transmitter localization problems.

None of the above spectrum sensing approaches address the privacy of the user who participates in the system, which is seen as one of the major issues limiting the deployment of cognitive radio networks [32]. One privacy vulnerability is that the RSS measurements (without the coordinate) can be used to locate a receiver [33], [34], for example using RSS fingerprinting methods like [26] or maximum likelihood estimators as in [14]. Cryptographic methods can help by limiting the resolution of RSS information provided to the central controller [34]. Our method modifies both the RSS measurements and the provided coordinates so that the attacker is unable to estimate the true receiver coordinates any better than the provided coordinates.

IX. Conclusion

We addressed the problem of location privacy in the context of crowdsourced localization of spectrum offenders where participating receivers report RSS measurements and their location to a central controller. We presented a novel adjusted measurement approach in which pseudo-locations are generated at random and are reported along with adjusted RSS measurements as if the measurements were made at the pseudo-locations. We used two RSS datasets, one from a cluttered office and another from roadways in Phoenix, Arizona to evaluate our approach. Our results show that location privacy can be preserved without a significant increase in the localization error. We also formulated an adversary attack that attempted to solve the inverse problem of determining the true locations of the receivers from their false locations. Our evaluations showed that the adversary does no better than random guessing of true locations in the monitored area.

We considered two adversarial situation. Primarily, we assumed that the central controller was an adversary and used local leaders to collect and adjust RSS measurements. However, our approach also applies to situations when the central controller is not an adversary but the applications using the collected data are adversarial.

References

[1] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.

[2] J. Krumm, "Inference attacks on location tracks," in *International Conference on Pervasive Computing*. Springer, 2007, pp. 127–143.

[3] J. Freudiger, R. Shokri, and J.-P. Hubaux, "Evaluating the privacy risk of location-based services," in *International conference on financial cryptography and data security*. Springer, 2011, pp. 31–46.

[4] E. Beatrix Cleff, "Privacy issues in mobile advertising," *International Review of Law Computers and Technology*, vol. 21, no. 3, pp. 225–236, 2007.

[5] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath, "The staircase mechanism in differential privacy," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1176–1184, Oct 2015.

[6] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," *Wireless Networks*, vol. 8, no. 2/3, pp. 187–197, 2002.

[7] R. Jose and N. Davies, "Scalable and flexible location-based services for ubiquitous information access," in *International Symposium on Handheld and Ubiquitous Computing*. Springer, 1999, pp. 52–66.

[8] M. Khaledi, M. Khaledi, S. Sarkar, S. Kasera, N. Patwari, K. Derr, and S. Ramirez, "Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2017, pp. 235–247.

[9] M. Bocca, O. Kaltiokallio, N. Patwari, and S. Venkatasubramanian, "Multiple target tracking with rf sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1787–1800, 2014.

[10] M. Bocca, O. Kaltiokallio, and N. Patwari, "Radio tomographic imaging for ambient assisted living," in *International Competition on Evaluating AAL Systems through Competitive Benchmarking*. Springer, 2012, pp. 108–130.

[11] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996, vol. 2.

[12] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," in *International Conference on Pervasive Computing*. Springer, 2009, pp. 390–397.

[13] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[14] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on signal processing*, vol. 51, no. 8, pp. 2137–2148, 2003.

[15] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Security and privacy (sp), 2011 ieee symposium on*. IEEE, 2011, pp. 247–262.

[16] J. Guo and N. Balon, "Vehicular ad hoc networks and dedicated short-range communication," *University of Michigan*, 2006.

[17] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Enhancing security and privacy in traffic-monitoring systems," *IEEE Pervasive Computing*, vol. 5, no. 4, pp. 38–46, 2006.

[18] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*. ACM, 2006, pp. 171–178.

[19] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *International Conference on Pervasive Computing*. Springer, 2005, pp. 152–170.

[20] B. Gedik and L. Liu, "A customizable k-anonymity model for protecting location privacy," Georgia Institute of Technology, Tech. Rep., 2004.

[21] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: Using social network as a side-channel," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 628–637.

[22] N. Vratonjic, K. Huguenin, V. Bindschaedler, and J.-P. Hubaux, "How others compromise your location privacy: The case of shared public ips at hotspots," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2013, pp. 123–142.

[23] R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux, "A distortion-based metric for location privacy," in *Proceedings of the 8th ACM workshop on Privacy in the electronic society*. ACM, 2009, pp. 21–30.

[24] R. Shokri, G. Theodorakopoulos, P. Papadimitratos, E. Kazemi, and J.-P. Hubaux, "Hiding in the mobile crowd: Locationprivacy through collaboration," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 3, pp. 266–279, 2014.

[25] D. J. Torrieri, "Statistical theory of passive location systems," *IEEE Transactions on Aerospace and Electronic Systems*, no. 2, pp. 183–198, 1984.

[26] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *19th Joint Conf. IEEE Computer and Communications Societies (INFOCOM 2000)*, vol. 2, 2000, pp. 775–784.

[27] J. Zhu and G. D. Durgin, "Extended indoor/outdoor location of cellular handsets based on received signal strength at greenville, sc," Georgia Institute of Technology, Tech. Rep., 2005.

[28] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*. IEEE, 2005, pp. 131–136.

[29] A. O. Nasif and B. L. Mark, "Collaborative opportunistic spectrum access in the presence of multiple transmitters," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. IEEE, 2008, pp. 1–5.

[30] R. K. Martin and R. Thomas, "Algorithms and bounds for estimating location, directionality, and environmental parameters of primary spectrum users," *IEEE Transactions on Wireless Communications*, vol. 8, no. 11, 2009.

[31] J. K. Nelson, M. R. Gupta, J. E. Almodovar, and W. H. Mortensen, "A quasi em method for estimating multiple transmitter locations," *IEEE Signal Processing Letters*, vol. 16, no. 5, pp. 354–357, 2009.

[32] M. Grissa, B. Hamdaoui, and A. A. Yavuza, "Location privacy in cognitive radio networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1726–1760, 2017.

[33] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *IEEE Intl. Conf. on Computer Communications (INFOCOM 2012)*, 2012, pp. 729–737.

[34] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Preserving the location privacy of secondary users in cooperative spectrum sensing," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 418–431, 2017.