

Privacy Preserving Localization Using a Distributed Particle Filtering Protocol

Tyler Ward, Joseph I. Choi, Kevin Butler, John M. Shea, Patrick Traynor, and Tan F. Wong
University of Florida, Gainesville, FL

Abstract—Cooperative spectrum sensing is often necessary in cognitive radios systems to localize a transmitter by fusing the measurements from multiple sensing radios. However, revealing spectrum sensing information also generally leaks information about the location of the radio that made those measurements. We propose a protocol for performing cooperative spectrum sensing while preserving the privacy of the sensing radios. In this protocol, radios fuse sensing information through a distributed particle filter based on a tree structure. All sensing information is encrypted using public-key cryptography, and one of the radios serves as an anonymizer, whose role is to break the connection between the sensing radios and the public keys they use. We consider a semi-honest (honest-but-curious) adversary model in which there is at most a single adversary that is internal to the sensing network and complies with the specified protocol but wishes to determine information about the other participants. Under this scenario, an adversary may learn the sensing information of some of the radios, but it does not have any way to tie that information to a particular radio’s identity. We test the performance of our proposed distributed, tree-based particle filter using physical measurements of FM broadcast stations.

I. INTRODUCTION

In cognitive radio networks (CRNs), collaborative spectrum sensing is often used to localize transmitters (cf. [1], [2]). There is an extensive literature on localization techniques for RF sources, involving various hardware, algorithms, environments, and topologies; cf. [3]–[7]. Received signal strength (RSS)/energy detection (ED) is commonly used, as it is easy to implement and is supported by most hardware.

Collaborative sensing systems are susceptible to a variety of attack vectors on the security and privacy of the participants’ information [8], [9]. Private location information may be leaked directly by the participants’ sensing disclosures or through variations in the aggregated results when a node joins a network. The case of multiple service providers collaborating to learn private information about a radio was introduced in [10]. Ref. [11] considers inference of a cognitive radio’s location based on his spectrum usage, which is reported to a database. Ref. [12] considers a more general approach that relies on a trusted third party providing the means for authentication, security, and privacy.

In this paper, we focus on the privacy of the locations of the spectrum-sensing radios (SSRs). Information exchanged by SSRs explicitly or implicitly carries the radios’ location information; however, in many scenarios, radios may wish to keep their locations private. This problem is easily solved if

there is a trusted fusion center. In [8], [9], a Privacy-Preserving collaborative Spectrum Sensing (PPSS) technique is developed that achieves privacy even when the fusion center is untrusted, if a trusted third-party is available. It uses a secret-sharing technique to allow a fusion center to recover the aggregate received energy reported by a group of sensors without either the fusion center or an eavesdropper being able to determine any individual sensor’s energy report.

In this paper, we also consider a PPSS scheme, but we consider the harder problem of *localizing* an RF source in a distributed network *without a fusion center or trusted third party*, while preserving the location privacy of the sensing radios. In the absence of a fusion center, we want to distribute the computation among the participating sensing radios, so that every radio has a similar computational burden. The desire to achieve privacy in the absence of any trusted party also leads us to partition the computation across the radios and ensure that any radio can only access a small portion of the total set of radio measurements. Moreover, we want to use a localization technique that can work with noisy measurement data and ambiguity about the channel path-loss exponent. Thus, we develop a protocol in which radios fuse sensing information through a distributed particle filter based on a tree structure.

We design our protocol to preserve privacy under the semi-honest (honest-but-curious) adversary model. In this model, there is at most a single adversary that is internal to the sensing network and complies with the specified protocol but wishes to determine information about the other participants. In the distributed particle filter, information about a radio’s signal measurement is passed to other radios through the *a posteriori* distribution of particles; this carries information about a sensor’s location. The protocol uses a combination of public-key encryption, an anonymizing proxy, and obfuscation packets to prevent an adversary from being able to associate sensing information with a particular radio’s identity, thereby achieving K -anonymity.

II. SYSTEM AND ADVERSARY MODEL

We consider a system of N radios that wish to perform collaborative spectrum sensing. The radios use peer-to-peer communication, and there is no trusted fusion center that can fuse the measurement information from the sensing radios. We consider the scenario in which the N radios are monitoring the emissions of a single transmitter that they wish to localize; however, the process discussed here easily generalizes to multiple transmitters in disjoint frequency bands.

The radios wish to perform localization by exchanging sensing information with their neighbors but do not trust the other radios with their location information. We assume that there is a single semi-honest (honest-but-curious) adversary, that wishes to analyze the content of the messages exchanged by the nodes in the fusion network to infer information about the location of any of the other nodes in the network. We assume that the adversary:

- is internal to the fusion network,
- may eavesdrop on all communications among nodes in the fusion network,
- may use physical-layer information to try to associate a node's transmission with its identity,
- does not know other nodes' private keys,
- carries out the protocol without manipulation or corruption, and
- does not inject false traffic into the fusion network.

These assumptions have several implications affecting the design of our privacy-preserving collaborative sensing protocol. Since the adversary obeys the protocol, it will not affect the result of the collaborative sensing process, other than through affecting how the sensor fusion may be performed while achieving the desired privacy goals. Since the adversary is internal to the network, we cannot transmit any location information directly because the adversary can use physical-layer information to associate that information with its transmitter. Finally, if the way information is distributed in the fusion process may reveal location information about the nodes, then existing node identities and public keys cannot be used.

III. LOCATION ESTIMATION

The location of the transmitter is estimated using a distributed particle filter algorithm. In this work, we assume that the radios have knowledge of the presence of the transmitter, along with its frequency, bandwidth, and transmission power, while the location of the transmitter and the path loss exponent of the propagation model are both unknown. The radios self-organize according to the protocol in Section IV, such that a subset of the SSRs will form a binary tree to fuse the sensing information. The use of only a subset of the SSRs is needed both to preserve privacy and to limit the maximum height of the tree, and thus the processing and information-exchange overhead. SSRs are sorted based on their signal-to-noise ratio (SNR), and nodes are chosen to populate the tree that have the highest SNR. The distributed particle filter at each node consists of three steps: input particle generation or collection, posterior probability calculation for each particle, and resampling. The steps vary slightly for the nodes depending on their location in the binary tree, as described below.

Leaf nodes initiate the particle-filter process by generating a random set of input particles. Each particle is assigned three values: latitude, longitude and path-loss exponent. The latitude and longitude are determined by generating points uniformly on a circle centered at the node's location. The path-loss exponent is generated by uniformly selecting values in [3, 4.5]. Internal (non-leaf) nodes receive a list of particles from each

child. The combined lists are treated as the input particles for these nodes.

Every SSR runs an update on its input particles based on its own RSS measurement. The attributes of each particle are used to calculate the value of the probability density function (pdf) of the total received power P_T , which is modeled as $P_T = P_I + X$, where P_I is the predicted received power of the FM signal if the transmitter is located at the particle's location, and X is an exponential random variable that models the noise power. The mean of X is set to $2\sigma^2$, where σ^2 is the noise variance of the in-phase and quadrature branches of the receiver. Specifically, the pdf of P_T is given by

$$f_{P_T}(p) = \frac{1}{2\sigma^2} \exp\left(-\frac{p - P_I}{2\sigma^2}\right) u(p - P_I), \quad (1)$$

where p is the RSS value measured by the sensor, and $u(\cdot)$ is the unit step function. The pdf value is set as the particle's weight for resampling using a bootstrap filter (cf. [13]).

Let $P_{i,j}$ be the number of input particles at node i in iteration j , and let F_j and G_j be constants that control the growth and attrition of particles, as defined in Section IV. At each non-root node at depth j , the bootstrap filter samples randomly from the input particles $G_j \cdot P_{i,j}$ times, but only uses the selected particle if it is among the top $F_j \cdot P_{i,j}$ particles with the largest weights. Then a new sample is generated using a Gaussian kernel centered at each particle selected by the bootstrap filter. The variances of the Gaussian kernel corresponding to the distance and path loss exponent are determined by linear interpolation using the node's SNR in decibels. By doing this, less trusted measurements allow resampling to explore a larger region of the sample space.

The root of the tree makes the final estimate of the transmitter's location. It partitions the range of latitude and longitude spanned by the final particle distribution into quadrants and selects the one with the most particles to be the new region under consideration. This process iterates recursively until reaching a prespecified precision. Then the location is chosen to be the center of the final region.

IV. PROTOCOL DESCRIPTION

We present a protocol to prevent a radio that receives particle information from associating that information with a particular radio in the distributed particle-filter process. To achieve this, all information is routed through an *anonymizer*, which is a radio in the fusion network that is elected to collect and distribute information to the other radios. The anonymizer does not carry out the particle-filter process to fuse information from other radios. All packets that pass through the anonymizer are encrypted using the public keys of the destination radios. We assume that, unless otherwise noted, packets do not contain headers or metadata that indicate which key was used to encrypt the data. Thus neither the anonymizer nor any node other than the intended recipient is able to read a packet and tell who it is for, and thus packets from the anonymizer must be broadcast or flooded to all the nodes in the network. We do assume that the packet contains a CRC or

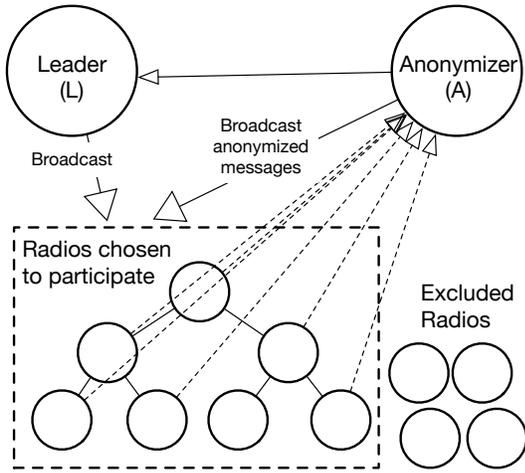


Fig. 1. The interaction diagram of the proposed privacy preserving distributed particle filtering protocol.

other information that allows the intended recipient to decrypt any packet that is intended for it and identify that fact.

We also provide techniques to prevent an adversary from gaining information by influencing the tree structure used in our distributed particle-filter process. We do this by electing a *leader* that is responsible for announcing the tree structure to be used in the distributed-particle filter process, where each vertex in the tree is associated with a public key of one of the radios carrying out the particle-filter process. A security analysis has been performed to show that, using this protocol, an adversary conforming to the threat model in Section II cannot discern any location information of any specific SSR during the protocol operations. The analysis is omitted due to lack of space. Our protocol is as follows:

Parameters and Notation:

The interacting participants of the protocol are summarized in Fig. 1 with the following list of parameters:

- N = number of participating radios
- F_j = proportion of “serving particles” at radio at depth j
- G_j = proportion of particles to resample at a radio at depth j
- P_0 = number of initial particles
- P_{\max} = maximum number of particles that can be sent
- D_{\max} = maximum depth of the tree
- RSS_i = received signal strength at radio i

Step 1: Initialization

At each radio i :

- 1a. Generate a random public key/private key pair (K_i, K_i^-) .
- 1b. Generate a random integer R_i uniformly on $\{0, 1, \dots, MN-1\}$, where M is chosen to reduce the probability of collision among random numbers.
- 1c. Generate a uniform field of $P_{i,0}$ particles for input to the first iteration (see step 4a below).

- 1d. The variable D_j specifies at what depth in the tree processing is being carried out in round j . Set $j = 0$ and $D_j = \min(D_{\max}, \lfloor \log_2 N + 1 \rfloor)$.

Step 2: Selection of anonymizer and leader

- 2a. Every radio broadcasts or floods R_i to the entire network.

- 2b. Each radio computes¹

$$\tilde{R}_i = R_i + \sum_{i=0}^{N-1} R_i \pmod{MN}.$$

- 2c. If the smallest and second smallest \tilde{R}_i are not unique, then each radio selects a new random integer R_i uniformly on $\{0, 1, \dots, MN-1\}$ and returns to step 2a.
- 2d. The radio associated with the smallest \tilde{R}_i has been elected as the leader (**L**), and the radio with the second smallest \tilde{R}_i has been elected as the anonymizer (**A**).

- 2e. **L** and **A** flood their public keys to all the other radios in the network. **L**'s public key is denoted by K_L , and **A**'s public key is denoted by K_A .

Step 3: Distribution of keys and selection of tree structure

- 3a. Each radio i encrypts its public key K_i and RSS_i with **L**'s key and then **A**'s key, $K_A(K_L([K_i, RSS_i]))$ and transmits that to **A**.

- 3b. **A** decrypts each key it receives to obtain $K_L([K_i, RSS_i])$.

- 3c. Once **A** has received keys from all the fusion radios, it transmits the $K_L([K_i, RSS_i])$ to **L** in a random order.

- 3d. **L** decrypts the messages from **A** with K_L^- .

- 3e. **L** uses the set of RSS s to create a mapping from the received keys K_i to the vertices of a tree, up to maximum height of D_{\max} .²

- 3f. **L** publishes the tree structure with vertices indicated by the associated keys K_i and distributes it to all the radios.

- 3g. Every radio checks if its public key is in the tree published by **L**. If not, it does not participate in the distributed algorithm. Those included in the tree are called *fusion radios*.

Step 4: Execution of Particle Filter

- 4a. At iteration j , if a radio is not at depth D_j in the tree, then it skips to step 4g.

- 4b. If $j \neq 0$, radio i combines the particles it has received to give a total of $P_{i,j}$ particles in iteration j . Note that for $j = 0$, $P_{i,0}$ is the set of particles from step 1c.

- 4c. Each radio i updates the particle weights based on its RSS measurement. The top $F_j P_{i,j}$ particles are designated as “serving particles”.

- 4d. Radio i then resamples the particles. It iterates over $G_j P_{i,j}$ candidates according to the following algorithm:

- i. Randomly sample from the particles according to their *a posteriori* probabilities.

¹Provided at least one radio chooses a random integer uniform on $\{0, 1, \dots, MN-1\}$, then each random value will be uniform on that range.

²The RSS s could instead be encrypted with order preserving encryption to mask the actual RSS values, while achieving the same effect.

- ii. If the particle selected is a serving particle, then it is used as the center of a Gaussian kernel to generate a new particle; otherwise, no new particle is generated. (In this way, radios that have better *RSSs* will generate more particles than radios with low *RSSs*.)
- 4e. If the last iteration (at the root), go to step 5.
- 4f. The particles are packaged into a packet of fixed size as follows. Additional random particles are added to the end of the particles so that the packet always contains P_{\max} total particles. An indicator of the number of real particles is added to the packet.
- 4g. Radios that are not in the tree at depth D_j send *obfuscation packets* encrypted with their own public key. Obfuscation packets are of the same size used by radios that are part of the tree at depth D_j , but contain P_{\max} random particles and an indicator that there are 0 real particles. These packets are used to prevent a parent node from using physical-layer information to determine which radios are its children and to prevent the anonymizer from determining which nodes are active in the tree at a given level, which would reduce the ambiguity between radios and their public keys.
- 4h. The particles are then encrypted according to the public key of the radio that is its parent/destination in the tree (or with the radio's own public key for obfuscation packets), encrypted with K_A , marked for routing to **A**, and transmitted to **A**.³
- 4i. **A** decrypts the packets it receives using K_A^- .
- 4j. Once packets have been received from every radio, the order of the packets is randomized, and the packets are broadcast/flooded to all the fusion radios.
- 4k. If a radio belongs to the tree at depth $D_j - 1$, it will try to decode all particle messages with its private key K_i . If the particle message decodes correctly, then any valid particles are retained to be combined with other particles the radio has received.
- 4l. After the round's communication is completed, the round is updated by $j = j + 1$, and the depth is decremented, $D_j = D_j - 1$.
- 5c. **A** decrypts the received packets with K_A^- .
- 5d. Once **A** has received and decrypted packets from all fusion radios, it transmits them in random order to **L**.
- 5e. **L** decrypts all the packets using K_L^- and discards all the fake estimates.
- 5f. **L** floods the network with the estimated location of the transmitter, which may also be sent to any other nodes outside the fusion network that need this information.

Illustrative Example

To illustrate the protocol described above, consider an example in which $N \geq 9$ and a full binary tree with $D_{\max} = 2$ is to be constructed. That is, seven sensing radios are selected from a set of N radios to locate a single transmitter. Assuming that the leader **L** and anonymizer **A** have been selected according to Steps 1-3 of the protocol, the leader populates the tree with the seven radios with the highest SNR values among all the N radios participating in the protocol. The resulting binary tree structure is shown in Fig. 2. The vertices of the tree are depicted in the figure by images showing each radio node's posterior particle distribution. The arrows indicate the flow of data as particles are passed from one sensor's output to the next sensor's input. In the images, the transmitter is labeled Tx, fusion radio n is labeled Rx n , and particles are displayed using their latitude and longitude.

When the algorithm starts, $j = 0$, and for this tree $D_0 = 2$. Since all of the nodes at this depth are leaves, they generate particles as described in Step 1c. A sensor's measurement along with the particles' associated path-loss exponent and location provide sufficient information to calculate the values of the pdf in (1). The particles are then resampled using the bootstrap filter and weights detailed in Section III and passed along for the $(j + 1)$ th round of the algorithm.

Continuing with this example, each of the nodes at depth $D_1 = 1$, receive two sets of particles. Receiver Rx4 receives particles from Rx0 and Rx1, and Rx5 receives particles from Rx2 and Rx3. Receivers Rx4 and Rx5 process their inputs using their own measurements, and pass their outputs on to the root, Rx6, which generates the final position estimate as described in Section III.

Step 5: Final Position Estimation and Distribution

- 5a. The root uses the final set of particles to estimate the transmitter's position, adds a flag to indicate that this is the true estimate, encrypts the packet⁴ with K_L , encrypts it with K_A , and transmits it to **A**.
- 5b. The non-root nodes generate a fake estimate of the transmitter's position, add a flag to indicate that this is a false estimate, encrypt the packet with K_L , encrypt it with K_A , and transmit it to **A**.

³Note that the packet could be first signed with the private key of the radio that generated the information to provide the parent with an integrity guarantee. However, under the threat model considered in this paper, the radios will not violate the particle-filter protocol.

⁴The packet could be first signed with the root's private key to prove to **L** that it came from the root, but that is not necessary under the passive-adversary threat model considered in this paper.

V. EXPERIMENTAL RESULTS

We apply the privacy-preserving particle filtering protocol described in Section IV to a set of experimentally collected RSS measurements. An Ettus Research B210 USRP radio was used to collect the physical measurements. A monopole antenna was attached to the roof of a car along with a GlobalSat BU-353-S4 GPS receiver. A program then periodically saved outputs from both of these sensors for post processing. A few different routes were taken between Ocala, FL and Gainesville, FL with measurements being collected approximately every 45 seconds. All of the effective sensor locations fall within a region that is nearly 10 km by 60 km.

For each measurement, the B210 radio collected 50 ms of raw samples. The power spectral density of the whole FM band was estimated from the raw samples. Then the power in

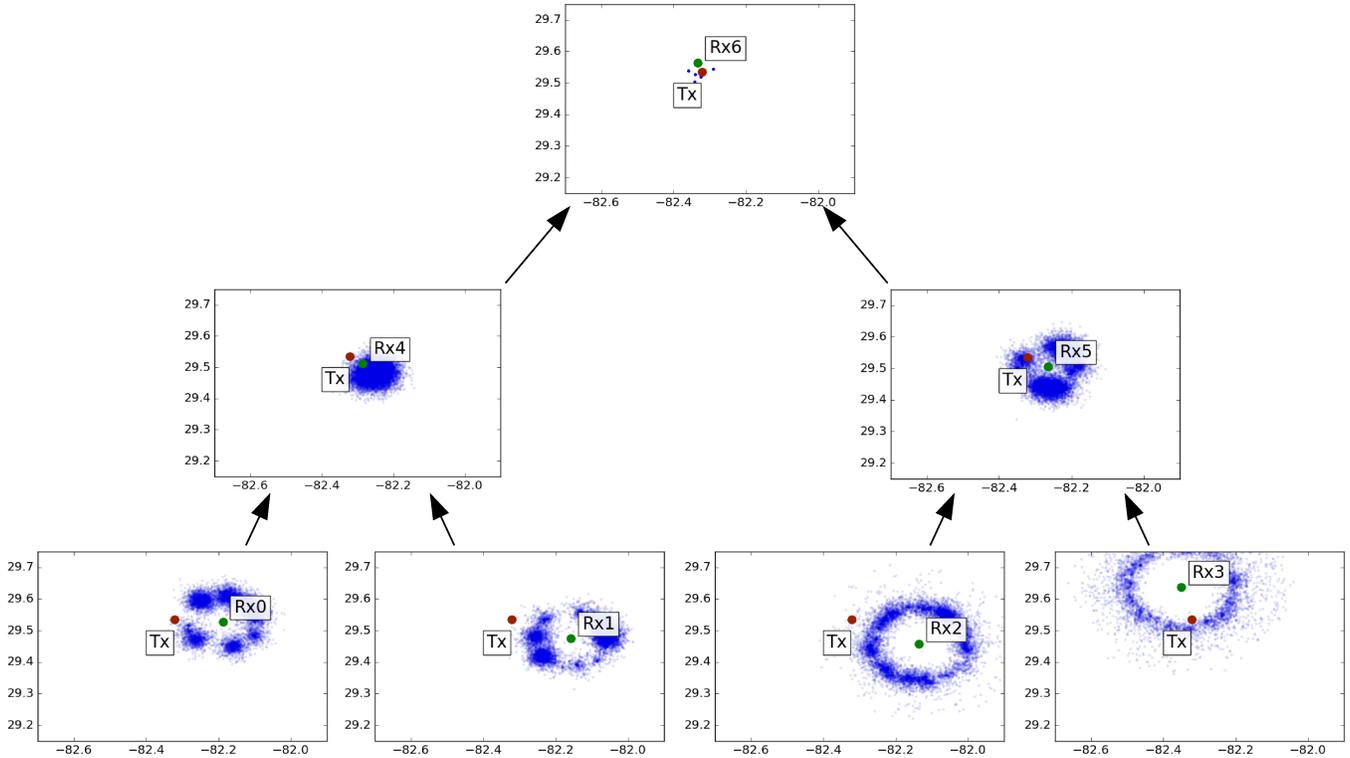


Fig. 2. This figure shows a typical flow of particles through the binary tree. Coordinates are in latitude and longitude, and the FM transmitter is labeled Tx while the receiver locations are given the labels Rx n .

a 200 kHz range corresponding to the 97.3 MHz FM broadcast station was determined. These values and their corresponding GPS coordinates became the measurements we used as inputs to the protocol. Altogether 100 *RSS* measurements were taken in the region mentioned above. These measurements were considered as measurements taken by 100 radios at different locations. The maximum tree depth in the protocol was set to $D_{\max} = 3$, and a complete binary tree with 15 nodes was employed. Particles were generated up to 75 km from the leaf node locations, and path loss exponents were generated uniformly between 3 and 4.5. This means the prior particle distribution's support, considering all of the leaves, was a total area of about 20,000 km².

The protocol described in Section IV was applied to $N = 30, 50,$ and 70 candidates randomly selected from the set of 100 physical measurements described above. These three choices model three different values of the number of radios in the sensor network. A thousand trials of the protocol were run for each case. In each trial run, the specified number of candidate nodes was randomly selected from the pool of physical measurements. The transmitter location estimate was obtained at the root, and the localization error was measured. Fig. 3 shows the root mean squared (RMS) localization errors for three cases as the number of initial particles generated at each leaf node varies from 1000 to 20000. We see from the figure that the RMS localization error drops as the number of initial particles increases. In addition, the RMS localization

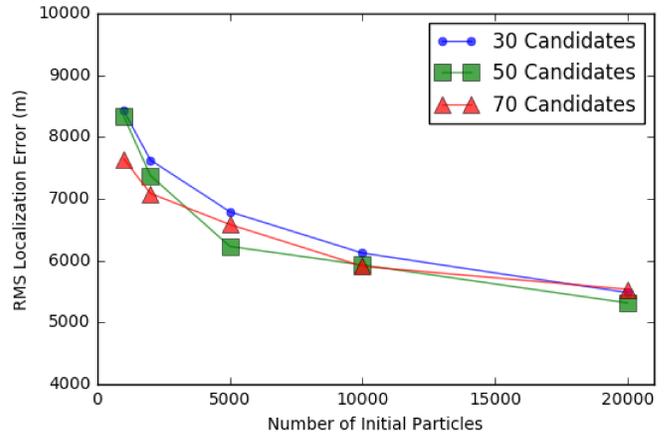


Fig. 3. This figure shows the RMS error for the location estimates of 1,000 trials and various values for the number of candidate nodes.

error decreases as N increases from 30 to 50. However, the RMS error may be higher or lower when N is increased to 70. This may be attributable to the highest RSS values clustering in certain locations, resulting in a decrease in spatial accuracy.

Figs 4–6 show the empirical distributions of the localization errors for the three cases of $N = 30, 50,$ and 70 with 20,000 initial particles. The most noteworthy trend is the difference in the tails of these distributions. A larger error tail seems more prevalent in the case of fewer candidate nodes ($N = 30$). This

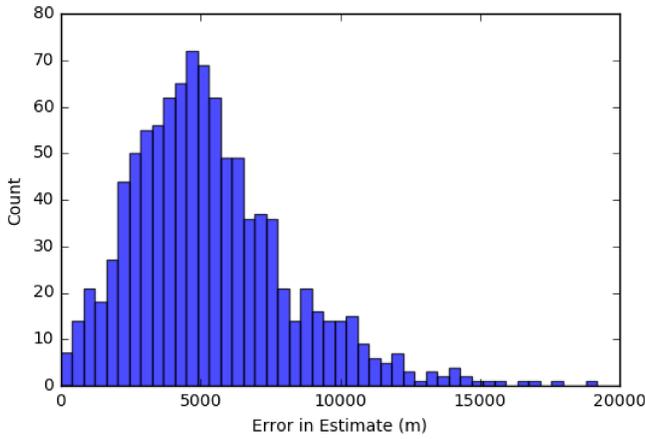


Fig. 4. A histogram of the error from 1000 estimates of a transmitter's location. The error refers to the distance between the estimated and true locations of the transmitter. Each estimate was generated using 30 candidate nodes from the experimental data set and 20,000 input particles per leaf.

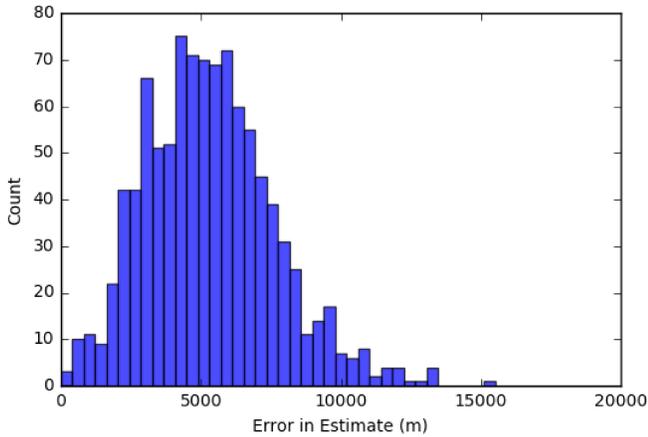


Fig. 5. A histogram of the error from 1000 estimates of a transmitter's location. The error refers to the distance between the estimated and true locations of the transmitter. Each estimate was generated using 50 candidate nodes from the experimental data set and 20,000 input particles per leaf.

may be attributable to some realizations having a collaborating set of candidate fusion radios that all have relatively weak RSS measurements; this is more likely to occur with smaller collaboration sets.

One of the likely contributors to the error in these results is model mismatch. Deficiencies in the model include shadowing and fading, which are not considered, and varying transmission power. The power of the downconverted FM signal is assumed constant by our model but in reality will vary depending on the content of the FM transmission.

VI. CONCLUSION

We present a distributed particle filtering protocol to localize a transmitter based on RSS measurements while protecting the location privacy of spectrum-sensing radios in a CRN. We design the protocol for the semi-honest threat model, in which an adversary faithfully executes the protocol but wishes to gain information on the location of the participants. Under these

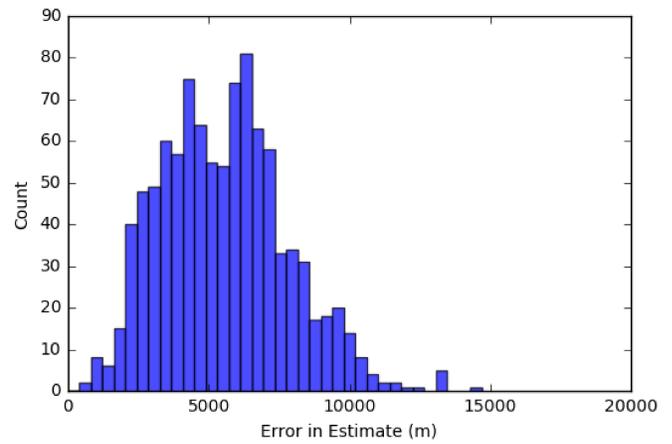


Fig. 6. A histogram of the error from 1000 estimates of a transmitter's location. The error refers to the distance between the estimated and true locations of the transmitter. Each estimate was generated using 70 candidate nodes from the experimental data set and 20,000 input particles per leaf.

assumptions, our protocol ensures that such an adversary is unable to gain information about the location of any specific participant during the protocol. Our protocol uses a novel distributed, tree-based particle filter, and we present results on the localization performance of this algorithm using power measurements from the FM band in North Central Florida.

REFERENCES

- [1] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview," *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 7, pp. 3386–3407, 2011.
- [2] B. Wang and K. Liu, "Advances in cognitive radio networks: A survey," *IEEE J. Select. Topics Signal Proc.*, vol. 5, no. 1, pp. 5–23, 2011.
- [3] E. Alimpertis, N. Fasarakis-Hilliard, and A. Bletsas, "Community rf sensing for source localization," *IEEE Wireless Communications Letters*, vol. 3, no. 4, pp. 393–396, 2014.
- [4] A. Konstantinidis, G. Chatzimilioudis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, and Y. Theodoridis, "Privacy-preserving indoor localization on smartphones," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 3042–3055, 2015.
- [5] J. K. Nelson, M. U. Hazen, and M. R. Gupta, "Global optimization for multiple transmitter localization," in *MILCOM 2006*, 2006, pp. 1–7.
- [6] A. Robertson, S. Kompella, J. Molnar, F. Fu, M. Dillon, and D. Perkins, "Distributed transmitter localization by power difference of arrival (PDOA) on a network of GNU Radio sensors," DTIC, Tech. Rep., 2015.
- [7] J. Wang, R. K. Ghosh, and S. K. Das, "A survey on sensor localization," *J. Contr. Theory and Appl.*, vol. 8, no. 1, pp. 2–11, 2010.
- [8] Z. Gao, H. Zhu, S. Li, S. Du, and X. Li, "Security and privacy of collaborative spectrum sensing in cognitive radio networks," *IEEE Wireless Communications*, vol. 19, no. 6, 2012.
- [9] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *Proc. INFOCOM*, 2012, pp. 729–737.
- [10] W. Wang and Q. Zhang, "Privacy-preserving collaborative spectrum sensing with multiple service providers," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 1011–1019, 2015.
- [11] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2751–2759.
- [12] H. Kim, "Privacy preserving security framework for cognitive radio networks," *IETE Technical Review*, vol. 30, no. 2, pp. 142–148, 2013.
- [13] T. Ward, E. L. Pasiliario, Jr., J. M. Shea, and T. F. Wong, "Autonomous navigation to an RF source in multipath environments," in *Proc. IEEE Military Commun. Conf.*, Baltimore, MD, Nov. 2016, pp. 198–203.