Crowdsourced measurements for device fingerprinting

Seth Andrews
Virginia Tech
setha@vt.edu

Ryan M. Gerdes
Virginia Tech
rgerdes@vt.edu

Ming Li
University of Arizona
lim@email.arizona.edu

ABSTRACT

Physical layer identification allows verifying a user's identity based on their transmitter hardware. In contrast with digital identifiers at higher protocol layers, physical layer identification or device fingerprinting can identify unique signal characteristics at the physical layer introduced by manufacturing variability specific to each device. Recently, dynamic spectrum access has been proposed to allow a larger number of devices to efficiently access wireless spectrum. In such a system many low-cost devices may be distributed over a large area with spectrum allocated and managed by a central authority. Traditional authentication methods may not be secure, or adequate to identify existing users in a backwards compatible way: Identifiers such as MAC addresses can be impersonated, and the number of devices and their distributed nature may make key distribution and revocation difficult. Consequently, physical layer identification can be used to augment other security measures.

We consider a crowdsourced scenario where individual users observe a signal using their own receiver and report their measurements to an enforcement authority which then identifies malicious users. Three types of measurements that can be crowdsourced are considered: actual signal observations, feature values, and fingerprinter output. Several methods for combining these measurements are considered. Performance is demonstrated on data collected from three wireless channels, used to simulate multiple receivers, from a total of twelve transmitters. The methods are evaluated in terms of required computational resources, bandwidth to report measurements, and how they are affected by mismatch in receiver characteristics. It is found that the crowdsourcing measurements can provide an improvement over individual receivers, with the best method dependent on the features and receivers used.

ACM Reference Format:
Seth Andrews, Ryan M. Gerdes, and Ming Li. 2019. Crowdsourced measurements for device fingerprinting. In *12th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '19), May 15–17, 2019, Miami, FL, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3317549.3323405

1 INTRODUCTION

As an increasing number of devices are capable of wireless transmission efficient usage of wireless spectrum is becoming ever more important. Dynamic spectrum access (DSA) networks will provide more efficient use of wireless spectrum by allowing wireless devices to cooperatively use spectrum. This includes primary users (PUs) such as radio or television stations which have existing rights to the spectrum, and secondary users which transmit on a non-interference basis. This requires that the DSA network determine when a channel is occupied, identify the user (particularly determining if it is the PU), and take action when an attack is detected. Crowdsourcing has been proposed to help monitor spectrum in such a system [6], including channel sensing [24, 28], and identifying the location of malicious users [12]. Incentive systems have been examined to encourage users to report measurements [15], and determine how to allocate a limited number of sensors to efficiently perform sensing [19].

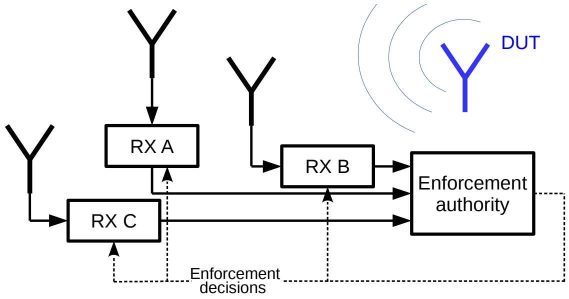

Figure 1: Diagram demonstrating a crowdsourced system. Three receivers capture observations of a signal and provide measurements (either the sampled signal or statistics extracted from it) to an enforcement authority to verify the transmitter's identity.

We consider the step of identifying malicious users in such a system, shown in Figure 1, using physical layer identification (PLI). PLI is a type of device fingerprinting which allows identifying a device based on characteristics and behavior unique to each device at the physical layer. This is introduced in each signal by manufacturing imperfections and variation in transmitter circuitry. One or more enforcement authorities fingerprint each device which transmits. Since these enforcement authorities may not be able to observe all users of the network, they rely on some devices reporting measurements. This results in a crowdsourced system where measurements from many low cost devices are combined for fingerprinting.

The contributions in this work include:

- Examining several ways of combining receiver measurements, which we classify in three levels based on where in the fingerprinting process the measurements are combined.
- Demonstrating a nonuniform reconstruction algorithm for combining multiple observations of bandpass signals.

WiSec '19, May 15–17, 2019, Miami, FL, USA

ACM ISBN 978-1-4503-6726-4/19/05…$15.00
https://doi.org/10.1145/3317549.3323405

- Discussing how mismatch in receiver characteristics will affect low level combinations of measurements, motivated by mismatch in interleaved analog to digital converters (ADCs).
- Show that crowdsourced measurements can provide better performance for transmitter identification than measurements from an individual receiver, under some conditions.

The paper is organized as follows: in the next section we describe a DSA network and outline a basic threat model. In Section 3 an overview is given of existing works on fingerprinting and crowdsourcing in DSA networks. Next, preliminaries are described including the basic steps to fingerprinting and three possible levels to combine crowdsourced measurements at. In Section 5 we define combinations used for each level, and discuss how mismatch between receivers can impact low level combinations. In Sections 6 and 7 the experimental setup and results are presented for these methods as well as performance without crowdsourcing. We conclude in Section 8, including some possible extensions to this work.

2 SYSTEM & THREAT MODEL

We describe a DSA network in more detail and the role fingerprinting can play in authenticating device identities. This is followed by a description of an attackers capabilities and objectives, and some limitations to the system model made to simplify analysis.

2.1 System

DSA networks allow re-using already licensed spectrum. A PU holds an existing license which they use only intermittently in time, space, or frequency. Secondary users are allowed to transmit opportunistically when the PU is absent. This allows for more efficient usage of existing spectrum, but introduces a number of challenges including reliably detecting the presence of the PU and identifying secondary users which misbehave. DSA networks are not tied to a specific technology, but exist alongside existing transmitters such as mobile telephone, television, or radar[13, 28].

We consider a DSA network, similar to that described in [6, 13], consisting of a central authority to manage spectrum allocations, individual users, and enforcement authorities. The central authority's responsibilities include allocating bandwidth and channels to individual users, changing allocations in response to reports of bad behavior, and preventing interference with the PU.

One or more enforcement authorities (enforcers) work to prevent abusive behavior. Misbehaving users have their allocation changed or their access blocked entirely, while users that are well behaved or helpful are rewarded with better spectrum allocations. There are not enough enforcers to observe every area covered by the DSA network, due to the cost and difficulty in deploying a large number of devices. Consequently, the enforcers rely on users reporting their observations of the physical layer to sense the channel and identify abusive users. The central authority can reward users who report measurements with additional bandwidth or more favorable allocations. This has been considered for spectrum sensing [19], here we extend it to allow device identification. This is similar to existing methods which use crowdsourced measurements of received signal strength (RSS) to identify a device's location [26], but our method links a device's identity to their transmitter hardware rather than location.

Each user is a device typically consisting of a transmitter and receiver, and may have some computational capabilities. These may include mobile phones, tablets, or wireless access points[12, 13]. Consequently, the users are not homogeneous: their receivers may operate at different sampling rates, have different quantization levels, and receive different levels of interference and fading.

In the following, we consider two types of users. The device under test (DUT) is a user whose identity we are interested in. Only a single DUT is considered at a time; any user transmitting may be the DUT. The DUT's transmitter is of interest, as this is what is fingerprinted. Secondly, we are interested in receivers. These are users with ability to observe the DUT's signal, and send measurements to an enforcement authority. The number of receivers reporting crowdsourced measurements may be small, both due to the limited receivers available and to reduce the overhead needed to report measurements.

2.2 Threat model

An attacker wishes to transmit without authorization. We consider attackers with hardware similar to that of legitimate users. Such low-end hardware is unable to record the physical layer observations of a signal with sufficient accuracy to impersonate other devices in a feature replay attack [5]. It is capable of recording and replaying a higher layer's information to steal digital identifiers.

Two attacks are considered. In the Sybil attack an attacker assumes multiple identities [23]. This may be a user of the system who wishes to avoid having misbehavior tied to their identity, or who has already been identified as malicious and banned. Fingerprinting at the physical layer can be used to identify this attack, and link the attacker to a known device. This can be done by verifying the DUT against each identity known to the enforcement authority, and taking likely matches. A related attack is the primary user emulation attack, where an attacker impersonates the PU. Since secondary users must not interfere with the PU, an impersonator has unrestricted access to the PU's spectrum. Fingerprinting methods have also been proposed to verify the PU's identity[18] and detect this attack.

We do not consider attackers attempting to corrupt crowdsourced measurements. Malicious devices working individually or in a group could send false measurements to mislead the enforcement authority. This is a legitimate concern, but outside the scope of this work.

3 RELATED WORK

Before our describing our approach to crowdsourced fingerprinting, we review a number of works related to the proposed method. First are works describing the current state of the art for PLI, and works that combine multiple measurements for fingerprinting. Last, uses of crowdsourced measurements in DSA networks are covered.

3.1 Fingerprinting works

A good overview of fingerprinting wireless devices at the physical layer is given in [5], as well as [23] which also covers fingerprinting methods using higher layers. These cover a number of scenarios where PLI is used in place of or to augment traditional identifiers. Most works use the same signal capture setup to gather reference and test data. Devices fingerprinted include RFID chips, WiFi, and

GSM. A variety of features have been used including power spectral density estimates, fast Fourier transform (FFT) coefficients, discrete wavelet transform coefficients, clock skew, and a variety of statistics extracted from the signal[5]. Features are extracted from a constant portion of the signal (such as synchronization symbols) or portions that contain arbitrary data. A number of works use multiple frames taken at different times, typically by taking the mean of a feature across all frames to reduce the signal to noise ratio (SNR) [4, 5].

Most works on fingerprinting (including this one) use high end hardware to observe signals. It is likely that some results will not hold when lower cost devices are used. In [18] fingerprinting is performed using low cost software defined radios (SDRs) as receivers. The performance of individual receivers varies substantially, although at a high SNR most provide acceptable performance for fingerprinting. Having a central authority with high end hardware (an oscilloscope) collect and distribute reference data to individual receivers is also examined, but it is found that this fails as fingerprints are specific to the receiver when low-end receivers are used. In [29] fingerprinting is examined in the context of identifying fake GSM base stations. However, using reference and training data from different transmitters is found to have no impact on performance. This shows that, in some cases, low-end hardware can be used in a fingerprinting system successfully. The same receivers (Ettus N210s) are used as in [18]), but the results are much better. This difference may be due to using a higher SNR (40dB, versus 15dB), different features, or other aspects of the experimental setup.

Using deep learning to identify cognitive radios is examined in [16]. Substantial pre-processing of each signal is undertaken prior to feeding it to a neural network. Signals are synchronized in time and frequency to provide the best performance. A neural network is trained to find the probability that the DUT generated the test data.Each frame is broken into segments, and the neural network's output for each segment is combined by multiplying the probabilities. In [2] several ways of combining measurements are given. Multiple frames are averaged to provide better reference data in training. In testing, multiple frames are used, but each is tested individually and the probabilities combined. This is combined with a committee of weighted classifiers, one classifier per feature. The methods are applied to multiple frames and features, rather than multiple observations of a single frame as in this paper.

3.2 Crowdsourcing measurements in DSA

A number of works have looked at crowdsourcing measurements in DSA networks. In most cases the objective is spectrum sensing. The nature of DSA requires that sensing be done securely [3]. A malicious user could manipulate decisions by reporting false sensor readings. Proposed solutions include more robust statistics, having a subset of known trusted users, and tracking each user's accuracy to create a per-user reputation [28] . An overview of spectrum sensing is given in [27], and includes some cooperative algorithms. Rules for combining crowdsourced observations are given. Most rules use hard decisions (a binary value indicating if the channel is occupied) although soft decisions (reporting a confidence level) have better performance with a small number of users.

In other works the objective is similar to fingerprinting, in that an attacker must be differentiated from a legitimate user. Crowdsourced measurements of RSS can be used to locate the source of a transmission[26]. These "location fingerprints" are not unique to each device, but rather a physical location. Attackers which move or are located close to a legitimate user may be misidentified.

4 PRELIMINARIES

Before examining crowdsourced fingeprinting, we first lay out the steps to perform fingerprinting and define several ways of combining crowdsourced observations of the DUT that could be used in a DSA network.

The following notation is used throughout the paper:

y	A signal, generated by the DUT and used to verify the identity of the DUT
$f(y)$	A function to extract features from a signal, y
T	Test data from the DUT, $T = f(y)$
R	The reference data for the DUT's asserted identity
$d(R,T)$	The distance between reference and test data
$V(y)$	The soft output of a fingerprinter, expressing confidence in the DUT's asserted identity
y^i	An observation of signal a signal from the DUT by receiver i, consisting of quantized signal levels
$C(\dots)$	A function combining multiple measurements of a signal, defined in the following sections

We assume that appropriate reference data is available. The reference data used in each scenario is described in the experimental setup, although that is not a focus of this paper.

4.1 Device fingerprinting

Fingerprinting can be used to identify the DUT or to verify the DUT's identity. Identification picks the most likely identities out of all seen by the fingerprinter, or determines that the device is unknown to the fingerprinter. Verification determines if the DUT's asserted identity is correct. We only consider verification, however identification can be performed by verifying against all known devices and picking the most likely identity or none at all. Verification is performed as follows:

(1) The DUT's asserted identity is extracted from the signal
(2) A set of reference features, R, known to come from that identity are taken from a database of reference data
(3) Test data is generated by extracting features from the observed signal, $T = f(y)$, typically using the identifier or another constant portion
(4) Reference and test data are compared, $d(R,T)$
(5) The DUT's identity is accepted if it falls within a predetermined threshold

These steps are depicted in black in Figure 2. A variety of features have been used, as mentioned in Section 3. Frequency based fingerprints have been found to perform well, and provide a large set of features with good performance [1, 4]. We use them in this work, and describe them fully in Section 6.1. The distance metric used to compare reference and test features can be chosen in a number of ways. Euclidean distance, cosine distance, or Mahalanobis distance all work well depending on the features used [1]. Each possible

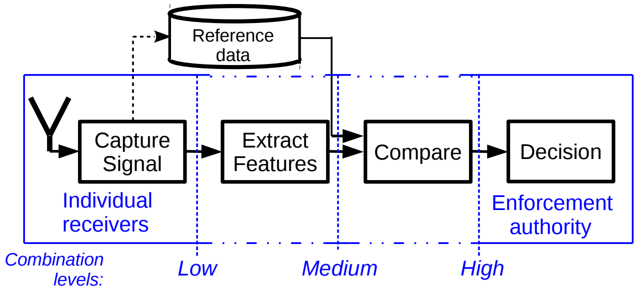

Figure 2: Steps in a typical fingerprinting system using a single receiver (black). Crowdsourced approaches are marked in blue, with the responsibilities of individual receivers and the enforcement authority shown. Different combination levels are labeled in italics.

threshold corresponds to a single true accept rate (TAR) and false accept rate (FAR).

The FAR describes how often the DUT is accepted when it is not the user it claims to be. This should be low for the system to keep out intruders. The TAR describes how often the verification system correctly identifies the DUT as a legitimate user, and should be high to allow honest users to access the system. The threshold provides a trade off between these two statistics.

4.2 Levels of crowdsourced measurements

We now consider crowdsourced measurements for fingerprinting. When multiple observations of a signal (or features extracted from it) exist they can be combined at several points in the fingerprinting process, which we designate

(1) **high** combining the outputs of each receiver independently verifying the DUT's identity
(2) **medium** combining the features extracted from the signal observed by each receiver
(3) **low** combining the sampled signal each receiver observes

These levels are depicted in blue in Figure 2. Clearly, each individual receiver must observe the signal from the DUT and the enforcer must make a final decision on the DUT's identity. High level allows each receiver to independently fingerprint the DUT using whatever methods they choose, the enforcer then combines these observations. Medium level combines multiple estimated features from each receiver. The low level combination requires the enforcer to combine signal observations from all receivers. Several options are available – we examine methods to combine the observations into a higher resolution signal. This signal can then be used with typical fingerprinting techniques.

The high and medium level methods are included primarily for reference, and are similar to existing methods used on multiple frames but applied to multiple observations of the same frame. We are most interested in low level combinations, covered last, which are based on combining multiple receivers' observations of a signal into a higher-resolution version. This allows each receiver to report arbitrary samples, including uniform samples with a rate below Nyquist and nonuniform subsets of the observed signal.

5 CROWDSOURCED MEASUREMENTS

We now examine each method. The performance of the different methods can be compared in several ways, including in terms of:

(1) overall performance
(2) bandwidth required to send measurements between receivers and the enforcement authority
(3) computational resources required at each receiver
(4) impact of mismatch on performance

Performance is evaluated fully in Section 7, and is evaluated in terms of the previously defined TAR and FAR. The bandwidth required to report observations should be minimized. By extracting more complex features less data can be sent. However, receivers may not perform substantial calculations due to computational or power constraints. Lastly, mismatch occurs when characteristics of receivers are not identical and their output is combined. Several types of mismatch related to interleaved ADCs have been analyzed.

(1) **Offset** occurs when DC offset of receivers is non-zero.
(2) **Gain** mismatch occurs when receivers exhibit a different range of gain, as will occur due to fading in the wireless channel or variation in amplifiers.
(3) **Timing** mismatch occurs due to differences in path length, unsynchronized receiver clocks, and independent noise in the triggering of each receiver.
(4) **Bandwidth** mismatch occurs when receivers exhibit different frequency response, due to frontend hardware and the channel used.

These primarily impact low level combinations. The processing done to extract features in high and medium levels can help correct for gain and offset mismatch, and minimize the effects of timing mismatch. As each level of combination is described the effects of mismatch are also given. Mismatch is discussed in more depth in Appendix A.1, including further solutions to mismatch.

5.1 High: combining fingerprinter outputs

In high level combinations each receiver acts as a fingerprinter or verifier, and the enforcement authority only combines the final confidence level of each receiver. This can be seen as similar to a committee of classifiers [2], but with each classifier consisting of a receiver independently sampling the signal, extracting features, and comparing these features to reference data. The measurement shared by each receiver can be hard or soft decisions. Hard is sharing a yes or no decision about the DUT's identity, while soft sharing uses a confidence level[28]. Additional steps need to be taken with soft combinations to limit the effect an untrusted receiver reporting false observations could have, although this is outside our system model. Hard decisions are naturally more robust to manipulation by a single user.

Soft decisions can be combined using the joint probability of all observations, found by multiplying outputs.

$$V_H(y) = C_H(y^1, ..., y^n) = \prod_{i=1}^{n} d(R, f(y^i)) \quad (1)$$

This is the same form used in [16] to combine classifier confidence in multiple subsections of a single frame. In practice, taking the mean of the log of the outputs provides greater numeric stability and allows easily changing the number of receivers reporting measurements. This method is very flexible. Receivers can use different feature sets as only the final verifier output is reported. Since each receiver's observation is processed independently the verifier's output is unaffected by offset or gain mismatch. It is also very low in terms of the amount of bandwidth required to report measurements to the enforcement authority.

Although not considered here, it is simpler for an attacker to manipulate by sending false measurements. Each receiver must have computational resources and reference data for any DUT. Although the medium and high level combinations would also work with the enforcement authority receiving signal observations and extracting features independently, this would negate the advantage of requiring less bandwidth.

5.2 Medium: combining features

Medium level combination combines the features extracted by each receiver. To do this, the mean of each feature over all observations is taken. Similar to many works which average multiple frames in time[5], this reduces the SNR as more observations are used. The resulting verifier is given by

$$C_M(y^1, ..., y^n) = \frac{1}{n}\sum_{i=1}^{n} f(y^i) \tag{2}$$

$$V_M(y) = d(R, C_M(y^1, ..., y^n)) \tag{3}$$

Other statistics, such as the median, could also be used. As with high level combinations, this is not affected by most types of mismatch. It requires medium overhead in terms of bandwidth, and provides some of the same flexibility as high level combinations. The number of receivers used can easily be changed, and the receivers can operate with different parameters as long as they are able to extract the same features. For some features, such as those based on the power spectral density or wavelet coefficients, this may require receivers operating at the same sampling rate. Receivers require some computational resources to extract features, but less than high level as they do not need to compare reference and test data.

5.3 Low: combining signal observations

Low level reconstruction uses observations from all receivers to attempt to reconstruct the original signal. The desired features can then be extracted from the reconstructed signal. By incorporating observations from multiple receivers the reconstructed signal has a higher sampling rate and should give better estimates of a feature's value. Designating this function as C_L, verifier output is

$$V_L(y) = d(R, f(C_L(y^1, ..., y^n))) \tag{4}$$

All processing is done at the enforcement authority so no processing capability is required at the receivers. The reconstructed signal has a uniform sample rate, making it easier to process. Although outside the scope of our attack model, it is more complex for an attacker to subvert. An attacker would need to know how features are extracted, what values other receivers have reported, and how their own reported measurements are used.

Of the combinations considered here, this has the highest bandwidth requirements, as the entire observed signal must be reported. The enforcement authority has more complex processing requirements and must account for mismatch between receivers, discussed in Section 5.3.3. Lastly, this method allows individual receivers to report signals with a sample rate below Nyquist. As long as the total samples from all receivers exceed the Nyquist rate the sample rate of any individual receiver is unimportant. The sample times still must be known approximately and each receiver must have sufficient bandwidth in the frontend hardware to accurately sample any signals in the frequencies of interest. Before elaborating on this method, two simple alternatives to handling low level combinations are given.

5.3.1 *Alternatives for low level data.*

Two simple approaches for combining the signals observed by each receiver are considered. Neither of these methods takes precautions to handle mismatch between receivers.

The first approach is to take the average across samples from each receiver, without correcting for timing or bandwidth mismatch.

$$C_L(y^1, ..., y^n) = \frac{1}{n}\sum_{i=1}^{n} f(y^i) \tag{5}$$

The desired features are then extracted from the resulting signal. Computationally, this is the simplest low level approach. However, it requires that the sampling rate of individual receivers be high enough to capture the signal without aliasing, negating one of the benefits of low level combinations.

The second alternative is to interleave samples from each signal and extract features from the higher resolution signal. We consider two approaches to correct for timing error.

(1) ordering the interleaved signals by start time to minimize timing error
(2) ignoring timing error and interleaving the samples with no regard to when each signal begins

Determining the ordering of observations introduces some complexity, compared to averaging samples. Neither approach takes into account mismatch between receivers as will occur in a realistic environment. This has a substantial negative impact on performance, as will be shown in Section 7.

5.3.2 *Nonuniform sampling algorithm.*

Nonuniform sampling algorithms provide an efficient and flexible approach to reconstruct the original signal[8]. Other approaches to handling nonuniformly sampled data are review in Appendix B. Given measurements from several receivers, $y^1, ..., y^k$, with corresponding sample times $t^1, ..., t^k$ we want to find an equivalent uniformly sampled signal. We designate the combined observations $\tilde{y}$, sampled at times $\tilde{t}$, so that $\tilde{t}_j$ represents the j th sample time out of all receivers and $\tilde{y_j}$ is the corresponding value (i.e. t_1 represents the first sample time out of all receivers with corresponding value $\tilde{y_1}$, similarly $\tilde{y}_r$ is the last sample taken at time $\tilde{t}_r$). The signal $\tilde{y}$ is bandlimited, to bandwidth B, since each receiver frontend contains a bandpass filter.

The frequencies, a, in a bandlimited signal y sampled at nonuniform times $\tilde{t}$ can be found by

$$a = T^{-1}b \in \mathbb{C}^{2M+1} \tag{6}$$

where M is the number of uniformly sampled frequencies to reconstruct within B, and

$$T_{l,k} = T_{l,-k} = \sum_{j=1}^{r} e^{-2\pi i(l-k)\tilde{t}_j}$$

$$b_k = \sum_{j=1}^{r} \tilde{y}_j e^{-2\pi i k \tilde{t}_j}$$

If frequency based features are desired the coefficients a can be used directly, removing the need to find the time domain signal. Otherwise, the value of y at time t is found by [8]

$$y(t) = C_L(t; y^1, ..., y^n) = \sum_{j=1}^{r} a_k e^{2\pi i k t} \tag{7}$$

This provides a uniformly sampled signal, which can then be used to extract fingerprinting features.

There are several things to note when solving (6) [8]. First, the Toeplitz structure of T allows for efficient solutions using iterative solvers, such as Levinson recursion. The dimension of T depends on the number of frequencies of interest, not on the number of sample points. This makes the solution computationally feasible even if a large number of samples are to be processed. Guarantees on convergence given in [9] are based on the maximum gap between sample times. These may not apply to bandpass reconstruction, but our empirical results show the method works well in most cases.

In implementing this algorithm some further points were discovered. Although a bandlimited formulation is given here following [8], it can easily be modified to handle bandpass data. This allows handling data at a sampling rate corresponding to the modulated data rate rather than the carrier frequency. This also significantly reduces computation time, as the majority of a are zeros when the sampling rate is significantly larger than the data rate. If the sampling geometry, $\tilde{t}$, is constant the Toeplitz matrix T and a substantial portion of b can be pre-computed, giving a much more efficient implementation. Consequently, changing the number of devices reporting signals, the sample rate of devices, or the total number of samples used will incur a substantial computational cost when using this approach.

5.3.3 Mismatch. Low level combinations are most impacted by mismatch between receivers, since measurements are combined directly without much of the pre-processing used to extract features in medium or high level combinations. The nonuniform sampling algorithm incorporates more timing information than the other approaches considered in this section, as it removes or minimizes timing mismatch. Both filtering and random interleaving can reduce the errors introduced by other types of mismatch, described further in Appendix A.1. Equation 6 incorporates both these solutions to mismatch. By finding a bandlimited signal, spurs introduced by mismatch are removed if they are outside the frequencies of interest. This also corrects for some distortion introduced in the band of interest rather than just discarding frequencies outside of it. Additionally, if the sample rates are not uniform across all devices this introduces a degree of pseudo-randomness similar to random interleaving. This does not remove all effects of errors, but it decreases the errors in the frequency domain. Consequently, this approach to low level combinations should perform well even with some mismatch present.

6 EXPERIMENTAL SETUP

The experimental setup is described, beginning with the frequency based features used, and how features are selected. The data collection setup is described, and the experiments presented in Section 7 are given. These include comparisons of low level methods, performance simulating several receivers without mismatch, and performance with actual mismatch present.

6.1 Subband frequency features

We use frequency based features, although the methods described do not depend on any specific type of feature. Two separate types of frequency fingerprints are used to present results. The low level combinations use the magnitude of frequencies of an irregularly sampled signal, using (6).

The high and medium combinations and the alternative low level methods in Section 5.3.1 use the log of the magnitude of FFT coefficients. Frequencies between $f_l = f_c - B/2 \leq f \leq f_c + B/2 = f_u$ are extracted as features, where f_c is the carrier frequency, and the signal has bandwidth B. Denoting the kth FFT coefficient of N sample points by F_k, a set of $\frac{NB}{F_s}$ features is given by

$$f(y) = \left\{ F_k(y) : \frac{f_l N}{F_s} \leq k \leq \frac{f_u N}{F_s} \right\} \tag{8}$$

Typically the sampling rate must satisfy $F_s > 2f_c$ to prevent aliasing. However, subband sampling is used with (8) to allow for lower sampling rates, comparable to what would be available in consumer hardware. A bandpass signal of bandwidth B can be accurately reconstructed if $F_s > 2B$ and $\frac{2f_u}{n} \leq F_s \leq \frac{2f_l}{n-1}$ for an integer n [22]. In this case, the frequency bounds become $\tilde{f}_l = f_l \ mod \ F_s$ and $\tilde{f}_u = f_u \ mod \ F_s$. In this case B should be related to the bandwidth of the receiver's frontend filter rather than signal bandwidth, so that other wireless signals do not alias into the bins of interest. While this is somewhat specific to our setup, since we acquire data modulated with the carrier from the oscilloscope, SDRs may operate in a similar manner by sampling to observe a large range of bandwidths rather than demodulating a signal at a specific frequency. An example of these features is shown in Figure 3, as well as the features extracted using (6).

6.2 Feature selection

There are a large number of possible feature sets when using radio frequency (RF) features. Including a large number features that do not distinguish well between transmitters will decrease performance. Reducing the number of features considered also improves computation time.

The Fisher criterion is a simple way to evaluate how well individual features can distinguish between transmitters. It is found as the ratio of average within class variance to total feature variance [11]. It is easily calculated, and can be made more resistant to outliers by using robust calculations of variance. The best rated features are then taken, shown in Figure 3, and the remainder discarded.

The distinguishability of FFT based features can also be evaluated based on amplitude. Frequencies with higher power give features

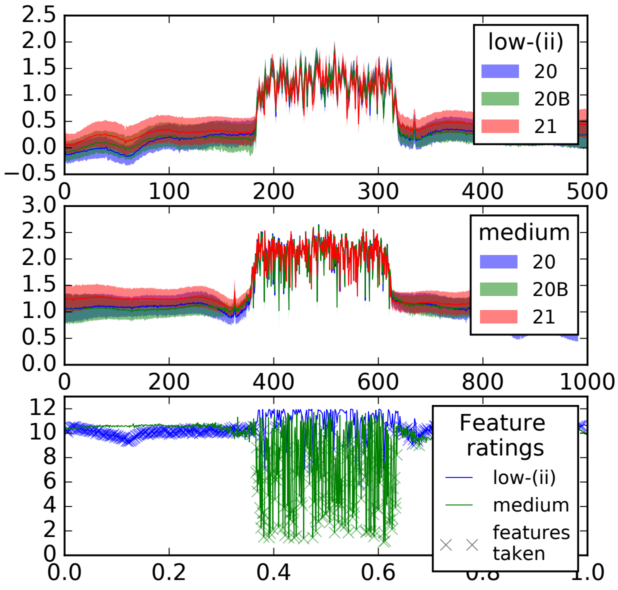

Figure 3: Example of features and feature selection. Upper: features extracted using Eq. (6), showing mean, quartiles shaded. Center: frequency features using Eq. (8). Lower: feature selection rating. There is some variation between the methods, and low level features favor features outside the main lobe.

with greater distinguishability. Fisher's criterion selects similar set of features, although it emphasizes features near the sidelobes of the signal's spectrum and ignores features nearest bins corresponding to the carrier.

6.3 Data gathered

We describe steps taken to collect data including transmitter setup, receiver setup, and processing. The sampling rate and number of samples are intentionally chosen to provide less than ideal performance for a single receiver, and demonstrate what improvements crowdsourcing allows for. Although results are shown for only a single sample rate and signal length the choice of these parameters has a substantial effect on the performance of crowdsourced methods as well as individual receivers. Further analysis is needed to determine the causes. The signals used in most of the experiments were decimated to a rate of 40 MHz, which is similar to that available in commercial off the shelf hardware (such as WiFi, which uses a bandwidth of 20MHz). However, there may be less noise and other advantages to the higher end hardware used.

6.3.1 Transmit setup. Ettus B210 radios [7] are used as the DUT. Each board has two transmit frontends, which we treat as separate transmitters. This provides a total of twelve transmitters. Each is connected in turn to a transmit antenna by an SMA cable, with the same antenna setup used for all transmitters.

The signal sent is generated in GNU Radio (version 3.7.10.1), on a computer running Ubuntu 14.04 LTS. The same bit sequence is sent in all frames, simulating a real scenario where an identifier or other constant portion of the message would be used for verification. The bit sequence was randomly chosen. This constant bit sequence is modulated using 4QAM with a bandwidth of 2.5 MHz and sent over the wireless channel at a carrier frequency of 2.422 GHz.

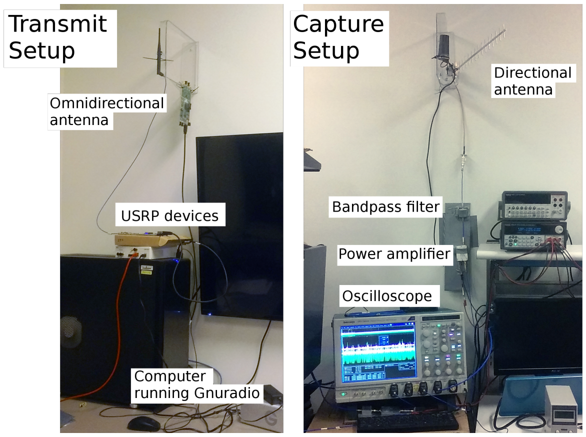

Figure 4: Data capture setup, showing transmitter and the oscilloscope with one receive antenna. Transmit and receive antennas are located on opposite sides of the lab.

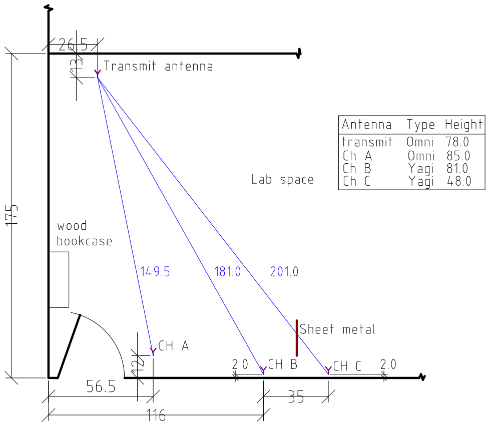

Figure 5: Diagram of antenna layout used. All dimensions are in inches, height is measured from floor. Tables, cabinets, other furniture and equipment below antennas are omitted.

6.3.2 Receiver setup. Data is collected in a wireless environment with other users transmitting. Three separate antennas connected to different channels of the same oscilloscope are used to simulate multiple receivers, shown in Figure 4. The layout of transmit and receive antennas is shown in Figure 5. The transmit antenna is separated from each receiver by 3.8 to 5.1 meters. The different channels introduces bandwidth and timing mismatch. Two of the antennas are placed above most foot traffic in the lab to avoid random shadowing. The third is placed 48″ above the floor with line-of-sight obstructed by sheet metal.

The received signals pass through a bandpass filter and amplifier before being sampled by the oscilloscope[20] at 25 GHz. Amplitude based triggering on one channel triggers all simulated receivers at the same time, however due to different signal paths a constant timing offset exists between channels. The filter covers the ISM band, so a significant amount of other wireless activity is still present. Each frame must pass several amplitude based checks to ensure that it is the signal we are interested in, and not another wireless signal. Running the capture setup when the DUT is not transmitting verifies that less than 1 % of the frames captured are unwanted. Each channel saves 5M samples when a valid frame is detected, and the capture setup is run until 1500 records have been collected from each transmitter.

The transient portion of each frame is discarded, so that the steady state portion is used for fingerprinting. The signal is decimated to simulate a lower sample rate. Before decimating, a random offset is chosen to simulate triggering with the lower rate ADC (e.g., the offset is randomly chosen as an integer in [0, D) samples, where D is the decimation factor). After downsampling, 2048 samples are taken from each frame, and normalized to have unit power. The normalization also partially fixes offset and gain mismatch.

6.3.3 Feature extraction and verifier setup. The following steps are common to all experiments, unless otherwise specified. The frames are split into a reference and test set for each DUT. A continuous set of 800 frames from the DUT is taken as reference, and the remaining frames from the DUT are used as the test set, as well as all frames from other transmitters. Each frame is decimated to have a sample rate of 41.6 M samples. Frequency features covering a total of 10 MHz are taken, totaling 983 bins.

The features in the reference data are rated using the Fisher criterion, and the top 250 are taken for reference and test data. The reference and test data are then compared using Mahalanobis distance. The TAR is found using the test data from the DUTs, and the FARs with the test data from all other devices.

6.4 Crowdsourced scenarios tested

Next, we describe several experiments to determine the performance of crowdsourcing methods.

6.4.1 Low level combinations. We consider five methods for low level combinations, based on those in Section 5.3.2 and 5.3.1.

i estimating frequencies using (6) when sample times are known exactly
ii using (6) with approximate sample times
iii averaging samples using (5)
iv the FFT of the interleaved signals ordered by start time
v taking the FFT of the signals interleaved without regard to start time

Method i uses the exact sampling times to create the matrix T in (5) for each frame observed. This introduces a separate sampling geometry for every frame observed, which requires recomputing T for each frame observed. This is very costly. Method ii uses the same algorithm but T is created with uniform sample times. The reported samples are ordered to approximate the uniform times. This removes some — but not all — timing error. This same technique is used to find approximate times before interleaving in method iv. Methods iii-v use the frequency features described in Section 6.1 once the combined measurements are found.

Three receivers are simulated using data from channel B. This creates data with only timing mismatch. Any algorithms that perform poorly under these conditions are not worth pursuing. The number of DUTs and frames processed for each has been reduced due to the long computation time required for method i. The frames are then split into reference and test data, and the reconstructed frequencies used as features to find verifier performance.

6.4.2 Crowdsourced, no mismatch. Similar to the previous section, performance without mismatch is found for all crowdsourced methods. Due to the random offset done before decimation, each receiver acts as though it triggers independently. This introduces random timing mismatch in each frame due to triggering, but not due to any specific device. Bandwidth, gain, and offset mismatch are not present, which is similar to the situation where channel equalization is performed.

The low level combinations are used as described in the previous section. For medium level the features extracted from each receiver's signal are averaged, then the frames are split into reference and test data.

The high level combinations have each receiver operating as a verifier. For each receiver, the signal is taken, features are extracted, and all frames are split into reference and test data. A comparison is made, and the distance of that frame from the receiver's reference data is returned for each receiver.

6.4.3 Crowdsourced, with mismatch. The combinations are handled as in the previous section. We use observations from the three receivers described in Section 6.3.2. This introduces timing mismatch due to the different path lengths in each the channel, bandwidth mismatch due to different channels and receiver setups, offset mismatch, and gain mismatch due to different amplifiers.

7 PERFORMANCE

Performance is shown for individual receivers as well as crowdsourced methods. The individual receivers are presented first, to establish a useful baseline for crowdsourced performance. The low level combination are presented next to find the best approach to compare with the other crowdsourced combination levels. After this results with only timing mismatch are presented, followed by results where the receivers have timing, gain, offset, and bandwidth mismatch.

Performance is described using the true accept rate (TAR) and false accept rate (FAR), defined in Section 4.1. The TAR describes the percentage of time that a legitimate DUT's identity is correctly verified. The FAR is the percentage when an attacker using a false identity is not detected. The desired TAR is near one and the desired FAR is close to zero. These statistics are related by the threshold which the verifier uses. As the threshold increases the TAR increases, at the expense of a corresponding increase in the FAR. This trade-off can be visualized using receiver operating characteristics. Receiver operating characteristics are found by taking TAR, FAR pairs for all thresholds. Each point on the curve corresponds to a particular threshold, and indicates the system's performance when using that threshold. Some works also use the equal error rate, which is found by choosing a threshold so that FAR and TAR are identical [5]. For our applications, having a high TAR is most

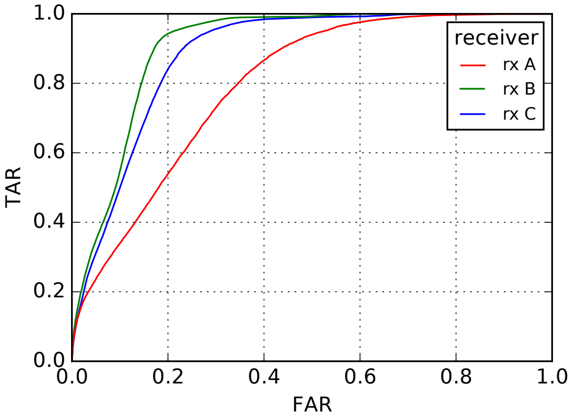

Figure 6: Performance of each individual receiver. The line of sight antennas perform best. The omnidirectional antenna results in a lower SNR and poorer performance.

important, as this measures the impact the verification system has on legitimate users.

7.1 Individual receiver performance

Before describing the crowdsourced experiments, it's useful to present results without using crowdsourced data, shown in Figure 6. Methods of combining crowdsourced measurements should provide better performance than any individual receiver. Otherwise, just that receiver could be used. Receiver B has the best performance (not surprising considering it uses a directional antenna with line-of-sight to the DUT). A TAR of 0.90 requires an FAR slightly over 0.15. Receiver C has a similar behavior for TARs below 0.5, but requires much larger FARs as the TAR approaches 1. Receiver A (which uses an omnidirectional antenna) exhibits the worst performance, with a FAR over 0.40 for a TAR of 0.9.

7.2 Crowdsourced performance

The results using crowdsourced combinations are now considered.

7.1 Low level combinations. There is considerable variation between the low level combinations even when there is no mismatch between receivers, shown in Figure 7. Interleaving the signal and taking the FFT (v) has the worst performance. This is regardless of whether the ordering is approximate or random. A TAR of 0.9 requires an FAR above 0.4. In contrast, the single receiver in Figure 6 can achieve a TAR over 0.95 while allowing an FAR of only 0.20. Averaging before performing the FFT (iii) provides better performance without incurring much computational complexity. It has a better FAR by 5 to 8 over a range of values, but none of the alternate methods can achieve a TAR over 0.8 and maintain a moderately low FAR.

Equation (5) arguably provides the best performance. When sample times are known exactly features determined with the nonuniform sampling algorithm give performance equal to or better than any individual receiver. Unfortunately, the exact sample times requires substantially more computation so they are not analyzed in subsequent sections. In the remainder of results we show performance for (ii) and (iii): nonuniform reconstruction with approximate times and taking the FFT of the averaged samples.

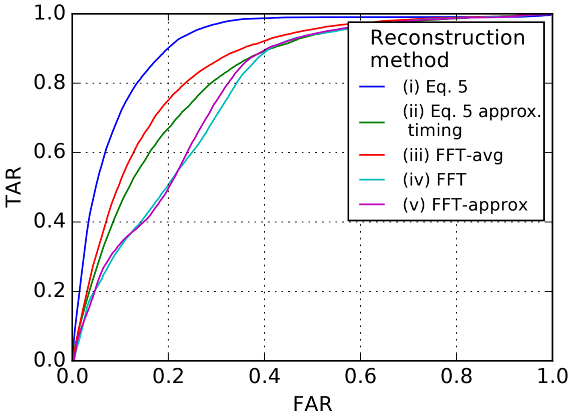

Figure 7: Performance low level methods, following the approaches outlined in Section 6.4.1. The approaches are based on whether sample times are known exactly or approximately, and whether the FFT directly or a nonuniform sampling algorithm is used.

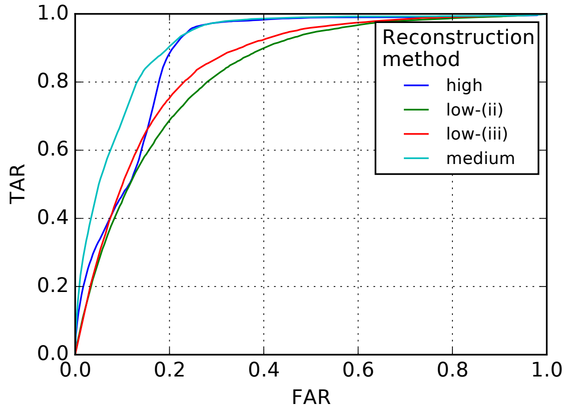

Figure 8: Performance of different combination methods with no mismatch. All methods perform well, with high and medium levels giving the best performance. Medium outperforms any individual receiver.

7.2 Crowdsourced, no mismatch. With no mismatch, shown in Figure 8, the medium level gives the best results, followed by high level combinations. Low level is comparable to high level for low FAR, but has much worse performance than any methods if a very high TAR is needed. The medium level outperforms receiver B, showing that it can have better performance than any individual receiver. High level closely matches receiver B, suggesting it is not heavily impacted by receivers with poor performance. Both low level combinations seem to be impacted by the poorly performing receivers.

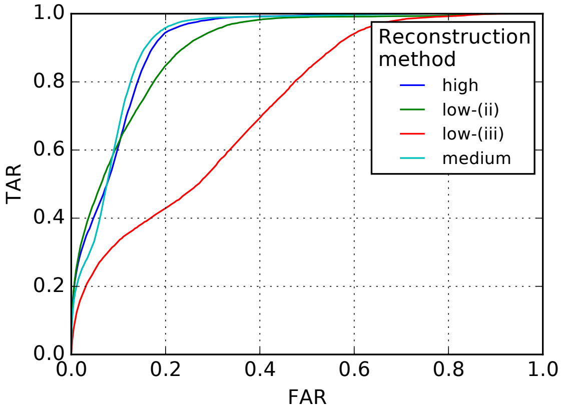

Figure 9: Performance of different combination methods with mismatch. Low-iii performs poorly, while low-ii improves in performance, for very low FARs it exceeds any individual receiver. Medium drops slightly in performance, but still outperforms any individual receiver.

7.3 Crowdsourced, all mismatch. With mismatch the results are substantially different, shown in Figure 9. The high level combination actually performs better than without mismatch, while the medium level has a slight drop in performance. Both these effects are probably due to the random variability in the choice of reference data rather than being related to the actual performance. Surprisingly, level (ii) outperforms the case with no mismatch as well. This is more than can be explained by random variation (and has been verified across multiple sets of reference data). However, low level (iii) has much worse performance. It would be almost unusable in most systems.

Further investigation is needed to determine under which conditions this holds, but under the conditions tested high, medium, and low level (ii) combinations can provide similar performance. Although it seems much simpler, the low (iii) method fails when there is mismatch in the receivers observations. In a practical system, this performance might be improved by performing channel equalization at each receiver to remove some mismatch.

7.3 Summary

In terms of performance there is not a clearly superior method when mismatch is present. Each may outperform others depending on the desired operating point on the receiver operating characteristic curves. Additionally, the sample rate and signal length were found to impact performance substantially, which is not analyzed here. For low FAR (under 0.10) the crowdsourced methods perform best, with very similar performance among them. However, an individual receiver can provide equal performance when the FAR is higher. The added complexity of low level methods did not show a substantial improvement in terms of performance, but it has other advantages discussed next.

Besides performance, other characteristics of interest are summarized in Table 1 for each method. Running time varies substantially between methods. Not surprisingly, the additional computations involved in low level combinations cause these to be the slowest of the methods. Low level (ii) takes five times longer, for three receivers, than the medium level. Out of the methods with acceptable performance characteristics, medium level combinations are the most efficient to compute closely followed by high level. In terms of bandwidth medium level combinations require substantially more than the high level, but not quite as much as low level. However, high level combinations require each receiver to store reference data for any DUT they need to verify. It may be possible to use less bandwidth with low level combinations by shortening the signal length sent, or reducing quantization levels. Further examination is needed to find the minimum number of samples required for good fingerprinting performance.

	computation at receiver (s)	computation by enforcer (s)	bandwidth per receiver (bytes)	minimum F_s per receiver
low-i	0.00	4208.71	4096	arbitrary
low-ii	0.00	157.21	4096	arbitraty
low-iii	0.00	17.16	4096	above Nyquist
low-iv	0.00	7.06	4096	above Nyquist
low-v	0.00	7.18	4096	above Nyquist
medium	9.60	0.54	1000	above Nyquist
high	11.38	0.02	4	above Nyquist

Table 1: Performance characteristics of different methods. Bandwidth assumes 4 bytes per feature and 1 byte per sample. Method low-v has been excluded as it has the same characteristics as low-iv. Low-level computations allow for a lower sampling rate and substantially less computation at the receiver at the cost of increased bandwidth for reporting.

8 CONCLUSIONS

A number of approaches to combining crowdsourced measurements have been examined. These can be used by an enforcement authority in a DSA network to securely verify transmitters identities. Advantages and downsides to all methods have been discussed. We have found that medium level combinations provide consistently good performance under most conditions. The low level methods investigated have varying performance, but we found that the nonuniform reconstruction with approximate timing information works well, exceeding medium level for some FARs when mismatch is present.

Further investigation is needed to determine under what conditions these methods will fail, and what types of data suite each best. Additionally, this could be extended to

(1) take advantage of spacial diversity by weighting channels according to SNR or other metric
(2) apply low level methods to an individual receiver using multiple frames, using observations in time instead of space
(3) examine low level combinations with observations from receivers operating at sub-Nyquist sampling frequencies

REFERENCES

[1] Seth Andrews, Ryan M Gerdes, and Ming Li. 2017. Towards physical layer identification of cognitive radio devices. In *Conference on Communications and Network Security (CNS)*. IEEE.

[2] Andrea Candore, Ovunc Kocabas, and Farinaz Koushanfar. 2009. Robust stable radiometric fingerprinting for wireless devices. In *International Workshop on Hardware-Oriented Security and Trust (HOST)*. IEEE, 43–49.

[3] T Charles Clancy and Nathan Goergen. 2008. Security in cognitive radio networks: Threats and mitigation. In *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications*. IEEE, 1–8.

[4] Boris Danev and Srdjan Capkun. 2009. Transient-based identification of wireless sensor nodes. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*. IEEE Computer Society, 25–36.

[5] Boris Danev, Davide Zanetti, and Srdjan Capkun. 2016. Types and Origins of Fingerprints. In *Digital Fingerprinting*, Cliff Wang, Ryan M Gerdes, Yong Guan, and Sneha Kumar Kasera (Eds.). Springer, Chapter 1, 5–29.

[6] Aveek Dutta and Mung Chiang. 2016. âĂIJSee Something, Say SomethingâĂİ Crowdsourced Enforcement of Spectrum Policies. *IEEE Transactions on Wireless Communications* 15, 1 (2016), 67–80.

[7] Ettus Research. 2017. USRP B200/B210 Specification Sheet. (2017). Datasheet.

[8] Hans G Feichtinger, Karlheinz Gr, and Thomas Strohmer. 1995. Efficient numerical methods in non-uniform sampling theory. *Numer. Math.* 69, 4 (1995), 423–440.

[9] Hans G Feichtinger and Karlheinz Gröchenig. 1994. Theory and practice of irregular sampling. In *Wavelets: mathematics and applications*. 305–363.

[10] Jeffrey A Fessler and Bradley P Sutton. 2003. Nonuniform fast Fourier transforms using min-max interpolation. *IEEE Trans. on Signal Process.* 51, 2 (2003), 560–574.

[11] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of machine learning research* 3, Mar (2003), 1157–1182.

[12] Mojgan Khaledi, Mehrdad Khaledi, Shamik Sarkar, Sneha Kasera, Neal Patwari, Kurt Derr, and Samuel Ramirez. 2017. Simultaneous Power-Based Localization of Transmitters for Crowdsourced Spectrum Monitoring. In *Proc. 23rd International Conf. on Mobile Computing and Networking*. ACM, 235–247.

[13] Vireshwar Kumar, He Li, Jung-Min Jerry Park, and Kaigui Bian. 2018. Enforcement in Spectrum Sharing: Crowd-sourced Blind Authentication of Co-channel Transmitters. In *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 1–10.

[14] Naoki Kurosawa, Haruo Kobayashi, Kaoru Maruyama, Hidetake Sugawara, and Kensuke Kobayashi. 2001. Explicit analysis of channel mismatch effects in time-interleaved ADC systems. *IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications* 48, 3 (2001), 261–271.

[15] Ming Li, Dejun Yang, Jian Lin, and Jian Tang. 2018. SpecWatch: A framework for adversarial spectrum monitoring with unknown statistics. *Computer Networks* 143 (2018), 176–190.

[16] Kevin Merchant, Shauna Revay, George Stantchev, and Bryan Nousain. 2018. Deep learning for RF device fingerprinting in cognitive communication networks. *IEEE Journal of Selected Topics in Signal Processing* 12, 1 (2018), 160–167.

[17] Sung-Won Park, Wei-Da Hao, and Chung S Leung. 2012. Reconstruction of uniformly sampled sequence from nonuniformly sampled transient sequence using symmetric extension. *IEEE Trans. on Signal Process.* 60, 3 (2012), 1498–1501.

[18] Saeed Ur Rehman, Kevin W Sowerby, and Colin Coghill. 2014. Radio-frequency fingerprinting for mitigating primary user emulation attack in low-end cognitive radios. *IET Communications* 8, 8 (2014), 1274–1284.

[19] Ahmed M Salama, Ming Li, and Dejun Yang. 2017. Optimal Crowdsourced Channel Monitoring in Cognitive Radio Networks. In *GLOBECOM*. IEEE, 1–6.

[20] Tektronix, Inc. 2017. DPO7000 Series Datasheet. (2017). Datasheet.

[21] Ralf van Otten. 2009. *Timing correction of time-interleaved ADCs*. Master's thesis. Eindhoven University of Technology, Eindhoven, Netherlands.

[22] Rodney G Vaughan, Neil L Scott, and D Rod White. 1991. The theory of bandpass sampling. *IEEE Transactions on Signal Processing* 39, 9 (1991), 1973–1984.

[23] Q. Xu, R. Zheng, W. Saad, and Z. Han. 2016. Device Fingerprinting in Wireless Networks: Challenges and Opportunities. *IEEE Communications Surveys and Tutorials* 18, 1 (Firstquarter 2016), 94–104.

[24] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. 2012. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *Proc. 18th International Conference on Mobile Computing and Networking*. ACM, 173–184.

[25] J Yen. 1956. On nonuniform sampling of bandwidth-limited signals. *IRE Transactions on circuit theory* 3, 4 (1956), 251–257.

[26] Rong Yu, Yan Zhang, Yi Liu, Stein Gjessing, and Mohsen Guizani. 2015. Securing cognitive radio networks against primary user emulation attacks. *IEEE Network* 29, 4 (2015), 68–74.

[27] Tevfik Yucek and Huseyin Arslan. 2009. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Comm. Surveys Tut.* 11, 1 (2009), 116–130.

[28] Rui Zhang, Jinxue Zhang, Yanchao Zhang, and Chi Zhang. 2013. Secure crowdsourcing-based cooperative pectrum sensing. In *2013 Proceedings IEEE INFOCOM*. IEEE, 2526–2534.

[29] Zhou Zhuang, Xiaoyu Ji, Taimin Zhang, Juchuan Zhang, Wenyuan Xu, Zhenhua Li, and Yunhao Liu. [n. d.]. FBSleuth: Fake Base Station Forensics via Radio Frequency Fingerprinting. In *Proc. 2018 Asia Conf. on Computer and Communications Security (ASIA CCS)*. ACM, 261–272.

A MISMATCH

Here the types of measurement mismatch are covered in more detail, followed by several approaches to removing mismatch. The difference between reconstructed signals with and without mismatch can be measured in terms of mean square error (MSE) in the time domain or spurious free dynamic range (SFDR) in the frequency domain. MSE measures the difference between samples in each signal, while SFDR expresses the largest difference in the frequency domain. Mismatch introduces spurs in the frequency domain which reduce the SFDR.

A.1 Sources of measurement mismatch

The effects of offset, gain, timing, and bandwidth mismatch are described, based on the models in [14].

(1) Offset mismatch occurs when a receiver has a non-zero DC offset. If b_i is the DC offset at receiver i, receiver i observes

$$y^i = y + b^i$$

If two receivers sampling at F_s are interleaved to form a signal with rate $2F_s$ it is as though a signal with frequency F_s had been added to the interleaved signal. In the frequency domain this introduces spikes at the DC frequency and F_s, the Nyquist frequency of the overall sampling rate [14].

(2) Gain mismatch is caused by ADCs at each receiver exhibiting a different range of gain. When wireless channels are used it may be caused by different amounts of fading in each channel. Receiver i with gain α^i will observe the signal as

$$y^i = \alpha^i y$$

For two receivers the ideal interleaved signal without mismatch is modulated by a signal of frequency F_s with power dependent on the amount of gain mismatch [14].

(3) Timing mismatch has three causes in the crowdsourced data: clocks at each receiver are not synchronized, path lengths from the DUT to each receiver vary, and independent noise in each receiver will cause each to trigger at different points in the signal. Consequently, the signal each receiver records is

$$y^i(t) = y(t + \delta^i)$$

where δ^i is not constant across frames. This also reduces the SFDR by introducing spurs in the frequency content. Consequently, timing mismatch will have a small effect on frequency based features if the spurs are introduced outside the features of interest.

(4) Bandwidth mismatch is introduced by different responses in the frontend of each receiver and each channel's frequency response. It is generally ignored in interleaved ADCs, as it can be minimized with careful choice of hardware and design of signal paths, but will be substantially larger in crowdsourced wireless data due to channel effects. With bandwidth mismatch the signal at an individual receiver is then

$$y^i = y * h^i$$

where h^i is the frequency response of the channel for receiver i and $*$ represents convolution.

Bandwidth mismatch has larger effects, and may affect some types of features (especially frequency based features). Consequently, it may have an effect on medium or high level combinations. The other types of mismatch are not a substantial issue for high and medium level combinations. Each receiver performs fingerprinting independently, by the time measurements are combined mismatch has been removed by forming an estimate of feature values or verifier probabilities. The effect of mismatch is largest on low level combinations, when samples are combined from all signals to create a higher resolution version. It can decrease the SFDR in the frequency domain, or MSE in the time domain.

While it is tempting to ignore the mismatch and accept it as part of each device's measurements this would make fingerprints dependent on the mismatch between users reporting and require that the same devices always report observations. Additionally, the effects of mismatch are dependent on the signal content, so unrelated wireless signals can change the distortion introduced by mismatch. It would be possible to discard observations from devices with very bad mismatch, but they should still have some information that can be used. For these reasons some correction must be made, particularly for low level combinations.

A.2 Solutions to mismatch

There are a number of approaches to remove effects of mismatch. Solutions developed for interleaved ADCs typically have minimal overhead and realtime operation. These constraints less applicable when processing crowdsourced data: some delay is acceptable, longer observations of a signal are available (rather than correcting each sample as it is generated) and more processing is possible.

Some approaches used in ADCs are not suitable, primarily

(1) using near-identical hardware and signal paths. With interleaved ADCs it is possible to design a system in this way, however in the crowdsourced case we cannot constrain users to have homegeneous hardware. Even if that were possible, sampling times are unsynchronized and the channel introduces timing mismatch and bandwidth mismatch.

Methods used in ADCs may also be partially suitable, including

(2) Calibrating based on a known signal [21]. This would be necessary for every DUT, and addresses offset, gain, and bandwidth mismatch. Calibration would be tied to a specific channel, causing issues if users move. Channel equalization may already be partially performed by receivers, partially implementing this. Since receivers trigger indepently this can not compensate for all timing mismatch.

(3) Filtering spikes and spurs introduced by mismatch. In the frequency domain mismatch introduces spurious images of the signal spectra. With a properly designed filter these can be removed, provided they do not lie in the frequencies of interest.

(4) Using a randomized sampling order [21]. Having ADCs sample in a random order removes the spurs in the frequency domain. The MSE is unchanged, but the error is spread out in frequency domain so that the SFDR is increased. This is adequate for our purposes, as long as no single frequency has large errors features based on frequency content should perform well. This occurs to some degree if receivers operate with different sampling rates, or could be achieved by receivers reporting a subset of their observations.

Approaches that are infeasible in ADCs are possible, primarily

(5) Normalizing signals to remove gain and offset mismatch. ADCs handle arbitrary signals, making this impossible. However, for PLI we are interested only in modulated signals. In fact, normalization is a typical step in most fingerprinting setups to ensure that signals have unit power and can be reasonably compared.

Approaches (4), (5), and (6) are part of the proposed low level reconstruction method, suggesting it can handle some mismatch.

B NONUNIFORM SAMPLING

We briefly review some approaches to handling nonuniformly sampled signals which are applicable to the low level combinations in Section 5.3. An early work is [25]. Several key cases for reconstructing a nonuniformly sampled bandlimited signal are given. Of interest here are arbitrary sampling sequences. Unfortunately the method given requires inverting a matrix whose size is related to the number of sample points, which makes all but very small problems impractical. A second case, periodic nonuniform sampling, occurs when several sample sequences with the same sample rate are interleaved with irregular offsets between sequences. This may be of interest in interleaved sequences, but requires all sequences have the same sampling rate.

Correcting the FFT of nonuniform data is considered in [17]. A transformation is found between the FFT of samples with uniform and nonuniform sample times. This also requires matrix inversion, limiting it to small problems. A "nonuniform FFT" has also been developed, see for example [10]. These methods generally take the FFT of samples treated as uniformly sampled, then apply a correction. However, we are not aware of any bounds on how much irregularity is allowable in the sampling sequence.

The approach chosen uses a reconstruction method based on the mathematical theory of frames, which allows finding a bandlimited signal given irregular sample points. It requires some constraints on signal bandwidth and the maximum difference between sample times [9]. The simplest solution is using Richardson's iteration, which allows sacrificing some accuracy for better computational times. Unfortunately this converges slowly and is very sensitive to the sampling geometry. Additionally, it requires reconstructing a bandlimited signal, rather than bandpass signal which results in significant computation for frequencies which are not present in a bandpass signal. Strohmer et. al improve on the iterative method in [8], which we describe in Section 5.3. This provides a direct formulation as a Toeplitz system. This has a faster rate of convergence, is less sensitive to inputs, and can handle a much larger number of samples than iterative solvers. It is also more computationally efficient and stable, and can take advantage of the additional structure provided by bandpass data.

ACKNOWLEDGMENTS

We are grateful to our colleagues at Virginia Tech who provided suggestions on the early stages of this work, and the reviewers and shepherd who provided additional useful feedback.

This work was partially supported by the National Science Foundation under grants CNS-1410000, CNS-1619728, and CNS-1731164