Extensions to Radio Frequency Fingerprinting

Seth Dixon Andrews

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Ryan M. Gerdes, Chair
Jeffrey H. Reed
Walid Saad
Ming Li

August 12, 2019
Arlington, Virginia

Keywords: Device Fingerprinting, Flux Capacitor, Transfer Learning, Physical Layer Identification

Extensions to Radio Frequency Fingerprinting

Seth Dixon Andrews

ABSTRACT

Radio frequency fingerprinting, a type of physical layer identification, allows identifying wireless transmitters based on their unique hardware. Every wireless transmitter has slight manufacturing variations and differences due to the layout of components. These are manifested as differences in the signal emitted by the device. A variety of techniques have been proposed for identifying transmitters, at the physical layer, based on these differences. This has been successfully demonstrated on a large variety of transmitters and other devices. However, some situations still pose challenges:

Some types of fingerprinting feature are very dependent on the modulated signal, especially features based on the frequency content of a signal. This means that changes in transmitter configuration such as bandwidth or modulation will prevent wireless fingerprinting. Such changes may occur frequently with cognitive radios, and in dynamic spectrum access networks. A method is proposed to transform features to be invariant with respect to changes in transmitter configuration. With the transformed features it is possible to re-identify devices with a high degree of certainty.

Next, improving performance with limited data by identifying devices using observations crowdsourced from multiple receivers is examined. Combinations of three types of observations are defined. These are combinations of fingerprinter output, features extracted from multiple signals, and raw observations of multiple signals. Performance is demonstrated, although the best method is dependent on the feature set. Other considerations are considered, including processing power and the amount of data needed.

Finally, drift in fingerprinting features caused by changes in temperature is examined. Drift results from gradual changes in the physical layer behavior of transmitters, and can have a substantial negative impact on fingerprinting. Even small changes in temperature are found to cause drift, with the oscillator as the primary source of this drift (and other variation) in the fingerprints used. Various methods are tested to compensate for these changes. It is shown that frequency based features not dependent on the carrier are unaffected by drift, but are not able to distinguish between devices. Several models are examined which can improve performance when drift is present.

Extensions to Radio Frequency Fingerprinting

Seth Dixon Andrews

GENERAL AUDIENCE ABSTRACT

Radio frequency fingerprinting allows uniquely identifying a transmitter based on characteristics of the signal it emits. In this dissertation several extensions to current fingerprinting techniques are given. These allow identification of transmitters which have changed the type of signal sent, identification when using different measurement types, and compensation for variation in a transmitter's behavior due to changes in temperature.

With thanks to my family for their support,
and to the many excellent teachers I've been blessed with over the years

ATTRIBUTION

The main chapters of this dissertation consist of manuscripts previously published, or in the process of review and preparation for publication. Consequently, these chapters contain some contributions from co-authors. These contributions are described now, as well as differences from the published version. Minor changes for formatting and to correct wording are not noted.

Chapter 2 is based on material published in the Proceedings of the IEEE Conference on Communications and Network Security (CNS), in 2017. It has been substantially revised and expanded for a journal publication. This includes a clearer description of when changes in bandwidth or modulation will change transmitter fingerprints, and further discussion of why. The experimental results have been expanded to include testing changes in carrier frequency and modulation types in addition to bandwidth; using data collected over a wireless channel; and refining the features used. Ryan Gerdes wrote substantial portions of the sections on the origin of similarity in fingerprints, and the security of invariant features. Both Ryan Gerdes and Ming Li contributed extensively to developing the experimental methodology and proofreading of the original manuscript. The section on configuration dependency was developed in response to comments from reviewers.

Chapter 3 was presented at the 12th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2019), and published in the conference proceedings. Ming Li is the principal investigator on the project, and contributed to the motivations and uses of the methods. Ryan Gerdes provided input on the early research ideas, primarily relating to low level combinations using methods based on interleaved analog to digital converters (ADCs). The writing is my own, as was the development of non-uniform sampling for low level combinations and high and medium level combinations as a comparison. Much of the final organization of the manuscript is due to helpful comments from the reviewers.

Chapter 4 is an unpublished manuscript, consisting of my own writing. The experimental design and analysis are my own. Ryan Gerdes and Ming Li provided input on the oscillator as a source of variation and the validation and use of regression models, as well as various editorial comments. They are co-authors on a submitted paper based on a shortened version of this chapter.

Lastly, the work in this dissertation has been partially supported by the National Science Foundation under grants CNS-1410000, CNS-1619728, and CNS-1731164.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Radio frequency fingerprinting consists of techniques for identifying a device based on its hardware, using features extracted from the frequency content of the signal created by it. Every transmitter has unique hardware - due both to design, and manufacturing variation between transmitters of the same model. This variation causes slight differences in the signal emitted by a wireless transmitter, which can be examined to track a device or verify the identify of a device seen previously. Features based on the radio frequency content of a wireless transmitter's signal are used. It is a type of device fingerprinting (or physical layer identification (PLI), since features of the physical layer are used to identify a device). Fingerprinting uses attributes that are unintentionally different between devices to identify them. Although fingerprinting can refer to identification techniques using attributes of the media access control or higher layers, in the following features extracted from the physical layer are used. Consequently, fingerprinting and PLI are referred to interchangeably.

Several extensions to current fingerprinting methods are considered, in the context of dynamic spectrum access (DSA) networks. DSA networks have been proposed to allow more efficient usage of existing wireless spectrum. Portions of the wireless spectrum are largely unused, either at certain times or in some locations. Other portions are over-used, and demands are increasing. DSA has been proposed to allow more efficient allocations of spectrum, including changing allocations in response to usage of the network and individual transmitter's requirements. This is partially enabled by software defined radios (SDRs), which capable of modifying a much greater range of configuration than traditional radio hardware. Modulation type, bandwidth, and other properties can be altered in software, responding to and shaping the environment. Identifying misbehaving transmitters is an important step enforce spectrum regulations in these networks. In contrast to traditional identifiers fingerprinting methods allow identifying devices without their cooperation(Table 1.1). Here, several challenges a DSA network poses to fingerprinting systems are considered. A more thorough description of DSA networks is given in Sections 2.2 and 3.2.

These challenges are described in the next section, as well as an overview of the remainder

Table 1.1: Comparison of traditional identifiers with fingerprinting methods for wireless devices. Points considered are based on the responsibilities of the device under test (DUT) and security in relation to attackers.

	Traditional identifiers	Wireless device fingerprinting
Responsibility	The device must provide an appropriate identifier	The fingerprinter must extract features to determine a device's identity
Cooperation	The device must cooperate	No cooperation needed from a device
Replay attacks	Message bit sequence can be recorded and replayed, in software	The device's signal can be recorded and replayed at the physical layer, using an oscilloscope and signal generator
Impersonation attacks	Encryption keys may be compromised, weaker identifiers can be copied	Behavior of transmitter's components can be imitated

of this section and the structure of the following chapters.

1.1 Structure of this work

Several extensions to current fingerprinting methods are examined. Chapters 2, 3, and 4 each consist of a self-contained manuscript describing a current challenge in fingerprinting wireless transmitters, reviewing existing literature, and describing and demonstrating proposed solutions.

In Chapter 2 the problem of fingerprinting transmitters capable of changing modulation type, bandwidth, or carrier frequency is considered. In a DSA network it is likely that SDRs will be used. This enables transmitters to change modulation type, carrier frequency, and the amount of bandwidth used in response to their own application requirements, and the requirements of other users of the spectrum. Unfortunately, such changes substantially alter commonly used RF features, and prevent fingerprinting. When a device has changed modulation type or bandwidth it becomes substantially more difficult, or impossible, to detect attacks such as the Sybil attack and impersonation attacks, described in Section 2.2.2. The proposed solution transforms features to be invariant with regards to bandwidth and modulation type. Results are demonstrated on a large collection of wireless transmitters.

Chapter 3 considers crowdsourced measurements for fingerprinting. As fingerprinting methods are deployed on a larger scale it will be advantageous to avoid deploying fingerprinting

hardware over a large area. A series of crowdsourced measurements can be used to identify devices by having some users of the wireless network report their observations of the network. This has been shown to offer some advantages for spectrum sensing, and here it is shown to aid fingerprinting performance as well. Several approaches are considered, and compared in terms of performance, receiver requirements, and processing demands.

Lastly, in Chapter 4 fingerprinting drift is considered. Drift occurs when fingerprinting features change gradually over time due to aging and changes in temperature or other environmental factors. Small amounts of drift degrade performance of a fingerprinting system, while large amounts of drift can cause a device to become unrecognizable. Here, only temperature driven drift is considered. It is shown that the oscillator is primarily responsible for this drift. Consequently, a more robust feature set which does not depend on the oscillator is investigated. Unfortunately these features, while more robust, are less capable of distinguishing between devices. Some approaches to model the fingerprinter's output to compensate for drift are examined, and found to work well.

The objective of the remainder of this section is to provide a general overview of applications and considerations for fingerprinting not specific to any of the problems being solved. Each chapter will go into more depth with a short literature review specific to the problem considered. Here, general considerations as well as points that apply to all chapters are covered, including current and past applications of PLI and related techniques. Practical considerations in a fingerprinting system are also covered. These include the fingerprinter's objectives, attacks that can be launched against PLI systems, types of features used, and distance and performance metrics used. This section concludes with a discussion of the contributions of this dissertation. The individual manuscripts follow, with some conclusions and possible future work.

1.2 A short overview of physical layer identification

A short overview of the development of fingerprinting is given, with a focus on current applications and techniques for fingerprinting wireless devices. This is followed by a number of areas fingerprinting has been applied to outside of communications. These include radio frequency identification (RFID) tags, hardware such as keyboards, and fingerprinting techniques for software. Lastly, some concepts similar to PLI are briefly reviewed. These include localization and watermarking, which use the physical layer for additional security but are distinct from fingerprinting.

Several surveys are available which go into more detail for some areas. A general overview of techniques for wireless transmitters and RFID tags is given in [2]. Methods specific to wireless transmission are given in [3], including features based on the media access control and application layers. Finally, [4] covers a variety hardware components that can be fingerprinted in mobile phones. These include cameras, microphones, and the RF frontend.

1.2.1 Wireless devices

The idea of tracking devices based on unique hardware characteristics began with specific emitter identification. This was developed for military applications, beginning in the 1960s for tracking targets [5]. The techniques were primarily developed by and for military users, and applied to radar and wireless devices [6]. Since then, a variety of uses have been proposed including in cognitive radio networks, and as an additional authentication mechanism for a variety of wireless devices. Most available publications begin in the early 1990s, and initially continued the focus on identifying wireless emitters, including devices causing interference [7] and radar [6].

More recently, fingerprinting methods have split between two objectives: identifying and authenticating devices. A natural application of identification is to detect the primary user emulation (PUE) attack in re-licensed spectrum. It has been proposed to sell secondary licenses to portions of the radio spectrum, on the condition that interference cannot occur with transmissions from the original user, or primary users (PUs). Consequently, a device that claims to be the PU can transmit without interference from secondary users. Fingerprinting methods were proposed as an early solution to securely identify the PU [8, 9]. Fingerprinting has also received some interest in digital forensics to tie transmissions to a specific device. If the device is recovered by law enforcement, it can be tied to previous illegal behavior [10]. Similarly, anonymous transmissions can be tied to other traffic from a device which contains information that may be tied to a real identity [11].

Authenticating (or verifying) a device's identity is another use of fingerprinting, and is compared to more typical authentication methods in Table 1.1. Characteristics of a device's hardware present in a signal can be used to help authenticate information in either a single message, or several messages from a device. An attacker can modify messages or copy identifiers, but cannot easily replicate a device's hardware. Sensor networks are examined in [12], including common attacks that fingerprinting can alleviate. The attacks considered focus on replaying messages, or the authentication portion of messages. These attacks will work on encrypted messages, since the content is replayed exactly. In this case, fingerprinting ensures the source of the message is legitimate, not just the content. Using fingerprinting as a step in authentication can also reduce network overhead, such as with cellphones where idle devices may consume substantial network resources. Fingerprinting has been proposed as a first step used to determine if full authentication should be performed for a device to accesses the network [13]. In this way, the total load is reduced. Fingerprinting methods can also be used when a protocol may not provide secure authentication. Some internet of things (IOT) devices may have very low processing capabilities, or low power consumption requirements. In these conditions, an adversary with much greater resources may be able to bypass the limited encryption algorithms they could employ [14]. Similarly, the CAN bus used by engine control units in automobiles has no authentication built in, and the current message format is too short to include authentication. Fingerprinting has been proposed to ensure that messages originate from an authorized device, without increasing the utilization

of the CAN network [15]. Lastly, WiFi access points are identified only by a network name, which can be easily discovered and copied. Fingerprinting would allow devices to ensure that they connect to an access point that has been used before, rather than a fake access point with the same name [16, 17].

1.2.2 Other applications of fingerprinting

Fingerprinting techniques have been applied in a number of areas besides wireless communications, some of which are briefly covered here. An area that has received substantial interest is authenticating RFID tags, due to their expected widespread adoption, low cost, and susceptibility to cloning. Fingerprinting has been demonstrated on a number of other objects, including keyboards and consumer electronics. Some fingerprinting techniques are also used in software on mobile phones and web browsers to track users.

RFID tags are small, cheap, passive electronic devices which can contain digital information. They can be embedded in documents to contain a digital version of the data, on cards to track access and usage, or attached to a variety of goods to track them in warehouses and other environments. However, information on an RFID tag can be copied to another tag. By using fingerprinting, these copies can be detected since the tag's physical layer behavior will change even when the information is copied exactly [18]. This allows detecting if a document with an embedded RFID tag is the original, and if goods are genuine or counterfeit [18, 19] (or rather, if the RFID tag attached to them is genuine). Unlike wireless transmitters, RFID tags are passive components and must be actively interrogated for fingerprinting.

Fingerprinting methods have also been applied to other types of objects. In [20] keyboards are fingerprinted to determine if a keylogger is present. The physical layer characteristics of a keyboard will change if a hardware key logger is inserted in between it and the computer. In this way, confidentiality of keystrokes can be ensured. It is possible to fingerprint any device with electronic components. This is similar to RFID tags, but the response comes from the actual electronics in the object being identified, rather than a tag attached to it. Fingerprinting was applied to laptops, cellphones, lightbulbs, and toy lightsabers, with the objective of eventually replacing barcodes at a cash register [21]. It is possible to perfectly identify an object's type, although identifying individual objects introduced some errors. Fingerprinting methods can also be applied to objects without electronics, such as identifying fraudulent checks by looking at features extracted from their paper [5]. Detecting counterfeit, recycled, or intentionally mislabeled integrated circuits is also mentioned in [4].

Lastly, a number of fingerprinting techniques have been used in software, including some specific to wireless devices. Attributes such as packet inter-arrival time, behavior of random back off timers, and the values of some fields defined in 802.11 frames can all be used to identify a device [3], either on their own or in concert with physical layer methods. Besides wireless devices, a substantial amount of effort has been put into tracking users online or through mobile devices. Properties such as screen resolution, timezone, system fonts, and

installed browser plugins can be used to identify the type of web browser visiting a website, the operating system used, and even track a specific user across multiple sites, Fingerprints have been collected from thousands of devices, and uniquely identify the majority of them [22]. On mobile phones a variety of sensors can be used. These include camera noise [23], accelerometer and gyroscope measurements[24], and magnetometer calibration data [25]. Although these are hardware components, they are accessed through software APIs. Consequently, they require some cooperation from the device being fingerprinted. Additionally, software can easily change the values of most of these properties, making this less suitable for authentication (although the assumption that hardware can't be easily modified does not always hold, as discussed in Section 1.3.2). These techniques are generally seen as a privacy concern, and vectors to exploit these features are being reduced or restricted. Even with better privacy and security controls in web browsers and on phones, some avenues for fingerprinting will likely remain.

1.2.3 Similar concepts

A few other methods use the physical layer for security. Some identify physical layer characteristics dependent on a device's location, rather than its hardware. While these use fingerprinting terminology, they are localization techniques which cannot distinguish between multiple transmitters. Others embed identifiers and other information in a transmission, using watermarking techniques. These are not discussed in depth, as they are traditional authentication and security techniques using a novel transmission medium.

Attributes of the physical layer that can be used to locate a device include received signal strength (RSS), and direction of arrival. Simulation shows that a network of devices reporting RSS values can identify an attacker as long as it is not placed near the device it is impersonating [8]. Using a network of devices makes it much more difficult for an attacker to successfully alter the RSS value at each receiver observing it. It would require approximate knowledge of the placement of all receivers, and multiple antennas. Channel characteristics are used in [26]. As with fingerprinting techniques, it may be possible for an attacker to imitate features from legitimate devices [27]. The features used in these techniques are unique to a location, and consequently can't distinguish between multiple devices in similar locations.

Watermarking embeds additional information in the physical layer of transmissions by slightly modifying the signal. Methods have been proposed that modify the frequency offset, cyclic prefix, and quadrature amplitude modulation (QAM) constellation [28, 29, 30]. Some variation in all of these properties can be corrected at the receiver, so this would not disrupt normal system operation. However, the signal to noise ratio (SNR) and the bit error rate will be impacted. Additionally, the information embedded at the physical layer is no more secure than any other data. Steps must be taken to secure the information in the embedded messages using cryptographic keys or other authentication methods. Consequently, these

methods have the same requirements and weaknesses as traditional authentication, while requiring the same fine grained analysis of the physical layer as fingerprinting.

1.3 Practical considerations

A number of applications of fingerprinting have been discussed. In this section several implementation aspects are considered. The overall objectives of a fingerprinting system include whether it is used for identification or authentication of devices; what performance should be obtained; and the number of devices that the system can handle. Another practical consideration is how a determined attacker may circumvent fingerprinting measures. Several attacks are described, of varying difficulty. This is followed by a description of features that can be used, and the equipment and techniques needed to extract them. The section concludes with a discussion of classifiers, distance metrics, and performance metrics that can be used with fingerprinting features.

1.3.1 Objective

When applying fingerprinting to a problem, several practical aspects and constraints must be considered. These include the overall objective of the fingerprinting system, and number of devices that can be successfully identified.

Fingerprinting can be used to perform authentication (verification) or identification (tracking). The objective will be determined by the application, but the features and classifiers used are largely the same. It can be assumed that devices are cooperative for authentication, and reference data can be enrolled by the users. With identification this may not be the case; devices may not be aware that they are being observed and tracked. In some cases only the class of device needs to be known, such as when identifying electronics at a cash register[21]. In most cases however, the objective is to track individual devices. This is largely determined by the use case, and features can be chosen based on this. Most of the literature focuses on individual devices, and demonstrates results differentiating between devices of the same type. This can be a harder problem, as individual devices of the same type often have very similar behavior. In the work presented here, authenticating or verifying a single device's identity is considered, using devices of the same type.

Related to performance (discussed in Section 1.4) is the capacity of a set of fingerprinting features. This describes how many devices can be differentiated from each other while maintaining a stated level of performance. This has some similar considerations to the attacks discussed in next section. Most works test a (relatively) small number of devices. As the number of devices increases overall performance will decrease. Practical applications may easily use hundreds or thousands of RFID tags or IOT devices, especially if the devices are mobile. Of the 17 papers covered in [2] the largest study uses 138 devices. Almost half

use ten or fewer devices. This suggests a possible disconnect between reported results and real world performance. Mutual information has been proposed predict performance as the number of devices increases [31]. This is also an issue when using neural networks. These have been used with fingerprinting methods, but generally learn features specific to each device they are trained on, and do not generalize well. In [32] it is found that the features learned by a convolutional neural network do not provide good performance for clustering when additional devices are introduced. In [33] it is shown that, while some data is needed from every transmitter to be identified, some stages of the neural network can be trained using data from only a few transmitters.

1.3.2 Attacks and security considerations

The assumption underlying PLI is that devices will have different behavior, and devices cannot easily change the behavior of hardware at the physical layer. These are not valid assumptions, and can be exploited by an attacker. Since features rely on random variation in hardware, it is quite possible an attacker may be able to acquire hardware with similar variation. For some types of features, it is possible to modify a device's behavior to mimic another device. With the use of an arbitrary waveform generator an attacker can recreate a device's signal (including the variation detected by fingerprints) with a high degree of accuracy. A similar issue is when a device alters its own fingerprint to avoid tracking or identification. In this case it may be possible to detect some attempts to modify a device's fingerprint [34].

An attacker can try to locate a device with hardware similar to that of a legitimate device. The difficulty is partially determined by the accuracy of a fingerprinting system, and is closely related to user capacity, discussed in the previous section. It also depends on the features extracted from the wireless signal, and the type and availability of devices. As mentioned previously, most works use a small number of devices due to cost and difficulty involved in testing. In a test involving 6000 RFID tags of 12 different models the classification error rate is comparable to other research, under 4% for 150 tags [19]. However, as the number of devices is increased the error rate also increases, ending at 45% when all 6000 tags are tested. However, only a single feature with a limited number of values was tested. At the very least, this shows that with a large number of devices and limited features finding one with similar behavior is very likely. For RFID tags this is a very feasible attack, depending on an attackers expected return. Transmitters are higher cost, but it still may be feasible for a sufficiently motivated attacker.

A bigger issue is that behavior of some hardware in a transmitter can be easily altered. In SDRs the sampling rates of the digital to analog converter (DAC), the carrier frequency, response of the frontend, and IQ offset can all be modified. With this, an attacker can create similar hardware, rather than searching for a needle in the haystack of dissimilar devices. This has been coined a feature replay attack, first demonstrated in [35]. It was shown that

by modifying an SDR to match a legitimate device's carrier frequency and constellation an attacker can fool the fingerprinting system over half the time. Features used are based on carrier frequency offset (CFO), IQ origin offset, and symbol errors. No correction was done for imperfections in the attackers hardware, beyond the CFO, so it is likely this could be improved.

Lastly, it is possible for an attacker to record and replay a legitimate device's signal, appropriately called the signal replay attack. This can be done with an oscilloscope and arbitrary waveform generator [36], or using an SDR[35]. It is more effective than the feature replay attack. In [36] found that an arbitrary waveform generator has a higher success rate with this attack. This limits it to a relatively advanced attacker who can acquire this hardware, and deploy it effectively. Transient based features (using the fast Fourier transform (FFT)) are shown to be more resistant to these attacks, most likely since they incorporate characteristics of the channel as well as the device [36]. An analysis of the expected cost to launch this attack is given in [37]. As SDRs gain more advanced capabilities and costs become lower it is likely that both the signal replay and feature replay attack will become even easier to launch.

These attacks are not examined in depth in the following chapters. Extensions to current fingerprinting methods are considered, rather than attacks on them. It has been suggested that modulation based features (especially CFO) are more vulnerable to impersonation attacks than features based on the transient or parts of signal spectrum [36]. With this suggestion in mind, RF features have been used as they should be more secure. Chapter 2 does discuss the resilience of the proposed techniques to signal or feature replay attacks. In a practical system these attacks should be considered as part of the attack model. In the applications considered require an attacker with considerable resources (an SDR or arbitrary waveform generator) compared to the devices (IOT, WiFi access points, or other lower-end transmitters) being secured.

1.3.3 Features

The features used will determine a system's performance, as well as the susceptibility to attacks discussed in the previous section. Features can be broadly categorized as predefined or inferred [2]., and the signal acquisition setups which allow acquiring these fingerprints are described. The effects of transmitter configuration on both types is discussed at further length in Section 2.1.1. Recently, some works have looked at automatically generating features using neural networks, and these are described, as well as features which allow fingerprinting using commercial off-the-shelf hardware.

Predefined features are based on specific, measurable properties of a signal such as CFO, IQ imbalance, or symbol error. Typical ranges or maximum values for these features may be listed in a data sheet or standards document. Many of these features can be easily extracted using a spectrum analyzer, or other specialized measurement equipment. These features

require greater knowledge of the signal type, but can be represented and stored in very little memory [38].

Inferred features are based on arbitrary transforms of a signal, such as properties of the signal envelope, or transient. They can be found in the frequency or time domain. These generally require having observations of the signal recorded with an oscilloscope or spectrum analyzer, although they have also been used with IQ data from a USRP receiver [9, 10, 31]. While these can provide a greater number of features with less effort, steps must be taken to choose the most effective features. A number of statistics that can rank features are examined in [39]. Most recently, using deep neural networks to learn features from a set of signals has been examined [32, 33]. These have shown good results, and may generalize to more devices and signal types with less effort than current techniques. These are grouped with inferred features, as it is unknown what attributes of the signal are learned by these networks.

Some devices also make physical layer information available in software. For instance, the 802.11 standard makes channel state information available, which can be used to infer CFO [40] or phase error in subcarriers [41]. Although not typical techniques used in physical layer identification, it is possible to infer properties of a device's hardware based on measurements higher in the protocol stack. The timing information in a large number of packets can be used to infer CFO [16]. This is encouraging, as it allows deploying fingerprinting systems for WiFi without having to design novel hardware or modify existing devices.

1.4 Distance and performance metrics

A large number of classifiers have been used for fingerprinting. As these are taken from the larger literature on machine learning they are not considered at length here, but details are available in [2, 10, 42]. Similarly, there are a number of metrics available for measuring performance in machine learning. Accuracy is frequently used to express the overall rate of classification errors, but this does not encapsulate the entire performance of a system. Here, results are presented in terms of the true accept rate (TAR) and false accept rate (FAR) of a fingerprinter. These statistics express the expected classification errors in terms of false positives (FAR) and false negatives (TAR). The classifier is based on a distance metric, a common approach taken in fingerprinting. In combination with the performance metrics this allows controlling the types of errors that occur, and provides an understandable meaning to the output. In combination with the receiver operating characteristics, it provides a clear picture of expected performance and trade-offs needed to achieve a given level of performance.

Several possible distance metrics will be described later, first the operation of a fingerprinting system operating in this manner is described. A fingerprinter requires several pieces of information. These are

R reference data from a know device

$\mathbf{D}$ a record from the DUT

d a distance or probability metric

τ a threshold to accept the device's identity, chosen based on the TAR and FAR

Reference data from the legitimate identity is compared to observations from the DUT, using an appropriate distance metric. The fingerprinter declares the DUT to be legitimate when

$$d(R, D_L) < \tau$$

and as an attacker when

$$d(R, D_A) > \tau$$

Increasing the threshold will increase the number of legitimate records accepted, but also increases the number of attacker's records incorrectly accepted. Likewise, decreasing the threshold decreases the percentage of times an attacker is accepted, known as the false accept rate (FAR). However, this also decreases the percentage a legitimate device is correctly accepted, known as the true accept rate (TAR). The TAR should be near 1, and the FAR near 0. In the next section, these performance metrics are discussed in further depth with other metrics. After that, various distance metrics used are discussed and compared, followed by a discussion of methods for normalizing the distance metrics.

1.4.1 Performance and receiver operating characteristics

Performance using TAR, FAR, and metrics based on these expresses performance for only a single possible operating point of the fingerprinter. Changing the threshold used by a fingerprinter changes the FAR and TAR, resulting in different fingerprinter performance. The set of FAR and TAR pairs for all possible thresholds creates the receiver operating characteristics. This allows determining the optimal performance (in terms of either the TAR or FAR) for a specific application. Classification accuracy confounds these two measurements, and does not allow for this trade off. What is acceptable performance will vary depending on the application. For detecting the PU in a cognitive radio (CR) network, impact on the PU must be minimized so a high TAR is more important. In most cases multiple records can be captured from a device, and used to make a decision. If only a single observation is to be used, the system must be robust to errors or overall system performance will need to be very high. In the following chapters, performance is considered for a single record. This works well to show the expected increase in performance that the proposed methods provide.

The best understanding of system performance comes from analyzing the receiver operating characteristics. However, many fingerprinting works summarize the performance of a system without the receiver operating characteristics. This can be done using the equal error rate (EER), using the area under the curve, or by choosing a fixed TAR or FAR. The EER finds

the point on the receiver operating characteristics for which the TAR and (1 - FAR) are equal. This is used frequently [2, 43], and is also used in presenting some of the results here. It is a simple summary of overall performance, but can obscure details. Measuring the area under the curve for the receiver operating characteristics provides a similar summary. Better performance leads to a larger area, but this still fails to capture details of specific operating points. Choosing a fixed TAR or FAR and observing the change in the other statistic gives the most detailed view of performance. The fixed TAR or FAR should be chosen appropriately for the application at hand, otherwise the change in performance is not meaningful.

1.4.2 Distance metrics

The best distance metric depends on the feature set used, and the amount of reference data available. The metric used can be chosen by comparing performance of several metrics, as was done when developing an invariant metric in Chapter 2, or it can be chosen based on knowledge of the features in use.

Euclidean distance

Euclidean (or l_2) distance is a simple distance metric, defined by

$$d_E(R, D) = \sqrt{\Sigma_{i=1}^{n}(D_i - \mu_i)^2} \tag{1.1}$$

Where μ_i is the mean value of feature i in R, and there are n features total. This treats all features as equally important. Larger features will have a bigger effect, specifically features where $D_i - \mu_i$ is large. Consequently, it generally should not be used where features have substantially different distributions.

Mahalanobis distance

Mahalanobis distance allows handling features with different variance, and has been used in other fingerprinting works [43]. It is used to establish performance of the features in the following chapters. The distance is found by

$$d_M(R, D) = (D - \mu)^T \Sigma (D - \mu) \tag{1.2}$$

where μ is contains the mean of each feature in R, and Σ is the covariance matrix of the features in R. It is proportional to the likelihood that the DUT's record originated from a normal distribution with the mean and covariance of the reference data. Consequently, it works best with features that are approximately normal. It can handle large differences in feature variance, as well as features that are highly correlated. However, it does require a

greater amount of reference data in order to accurately estimate Σ. It can continue to work well when a smaller amount of reference data is available, but over fits to the reference data and results in larger distances for legitimate devices.

Cosine distance

Cosine distance finds the angle between two sets of features, and is found by

$$d_C(R, D) = \frac{1}{|\mu||D|} \sum_{i=1}^{n} \mu_i D_i \tag{1.3}$$

where μ_i is the average value of feature i. The relation between features is considered, rather than overall changes in the feature set magnitude. Consequently, changes in magnitude of a single feature will be detected, but uniform scaling of all features is not detected by this metric. This distance metric works best with feature sets where the relation between features remains approximately constant, but the overall magnitude may change substantially between observations.

1.4.3 Normalization of distance metrics

The suitability of several distance metrics has been described for different data sets. Most distance metrics are very dependent on the reference data used. The amount of reference data, type of features used, and number of features used will all affect the fingerprinter output. It is useful to normalize the output of each of these metrics to remove these effects, and provide a consistent meaning for all reference data. This does not change the fingerprinter's performance – the interpreting distances using receiver operating characteristics doesn't assign any meaning to the distances. However, it will make it easier to understand what the distances indicate.

Two approaches to normalization considered in the following. First, normalizing distances by the standard deviation is considered, and second a function to limit distances to the range [0, 1] is given. A typical approach to normalizing distances is

$$\frac{|d - \mu|}{\sigma} \tag{1.4}$$

where μ and σ are the mean and standard deviation of the distances. These can be found for distances only from the current DUT, or using distances for all reference data available.

In some cases it may be useful to constrain the distances to a limited interval. This can be

done for the range [0, 1] using a sigmoid function

$$-1 + \frac{2}{1 + exp(-d)} \tag{1.5}$$

In the following chapters, normalization based on Eq. 1.4 is used. This was chosen over Eq. 1.5 as constraining the distances makes some changes less readily apparent. This includes Chapter 4 where the effect of changes in temperature on distances is examined. Eq. 1.4 allows clearly observing the effects, as the distances can change by large amounts without bounding. Similarly, Chapter 2 uses combinations of distance metrics to create invariant metrics. The invariant metrics will benefit from using a linear normalization. The data used to find μ and σ is indicated where normalized distances are used.

1.5 Contributions

Each following chapter considers a different challenge in wireless fingerprinting. The complete contributions are described with each chapter, but a brief overview of the key contributions is given here. In Chapter 2 fingerprinting transmitters which have changed configuration is considered. When transmitters change configuration, state-of-the-art fingerprinting features based on the FFT can fail to re-identify devices. Transmitters changing carrier frequency has been identified as a future problem in [44]. Changes in RF features due to modulation type are demonstrated in simulations in [45], but empirical results are not presented for multiple modulation types. Configuration dependency is thus an issue of potentially large interest in fingerprinting systems which has not been widely analyzed. Consequently, in Chapter 2:

- Configuration dependency is analyzed in terms of impacts on different features, and it is demonstrated that commonly used RF features fail to distinguish between wireless transmitters which change modulation type or bandwidth.
- A distance metric is developed which is invariant to changes in configuration from the DUT, and which can compare two sets of data containing different feature sets.
- The proposed invariant method is demonstrated on two different carrier frequencies, five modulation types, and a wider range of bandwidths. It is shown that it can successfully re-identify devices which have changed configuration, where state-of-the-art features fail.

Methods of combining different types of measurements for wireless fingerprinting are considered in Chapter 3. Crowdsourcing methods have been considered in DSA systems for spectrum sensing [46], and locating violators of spectrum policies [47]. Here, crowdsourcing methods are applied to device fingerprinting to allow identifying transmitters in a location-independent manner. Specifically,

- Several ways of combining receiver measurements are developed, and classified in three levels based on where in the fingerprinting process the measurements are combined.
- A nonuniform reconstruction algorithm for combining multiple observations of bandpass signals is applied and analyzed.
- The impact of mismatch in receiver characteristics, especially on low level combinations of measurements, is examined.

Lastly, in Chapter 4 the impact of fingerprinting drift on features is considered. Drift is the result of gradual changes in the features used for fingerprinting, and can have a substantial impact on a fingerprinter's performance. However, most research is limited to verifying that drift does not occur due to small changes in time or temperature [2]. In practical systems, drift is much more likely to occur. This was recently demonstrated in [48], using similar devices to those used here. Changes of 10 °C are sufficient to prevent identification of a device, when using common predefined features. The experiments here confirm similar results when using RF features, and are expanded to include examining various common transmitter components as possible sources of drift, and testing several methods to compensate for drift. It is found that

- The oscillator is primarily responsible for drift due to temperature.
- The CFO is a primary source of the ability to distinguish to between devices, for the RF features used.
- Models based on the behavior of transmitter components can compensate for drift in features.

Chapter 2

Fingerprinting after changes in transmitter configuration

Seth Andrews, Ryan M. Gerdes, and Ming Li

Cognitive radios have the ability to change modulation type, bandwidth and other configuration parameters in response to application requirements and the environment. Dynamic spectrum access networks take advantage of this ability to provide more flexible and efficient spectrum allocation. Physical layer identification, or device fingerprinting, has been proposed to help secure these networks by identifying wireless transmitters based on variation in hardware that creates signal characteristics unique to each device. Unfortunately, most features used for physical layer identification are negatively impacted by changes in transmitter configuration, diminishing their ability to identify legitimate users and attackers. In this paper we extend current physical layer identification methods to allow verifying the identity of wireless transmitters using features that vary with changes in configuration of bandwidth or modulation type.

We discuss the challenges changes in transmitter configuration present to fingerprinting, and develop a simple approach based on multi-view learning that allows

Expanded and revised from a paper published in Proceedings of the 2017 IEEE Conference on Communications and Network Security (CNS). Portions copyright 2017 IEEE.

This work expands on the method and results initially presented in [49]. We are grateful to the reviewers of this manuscript for their time and comments, as well as those who reviewed and commented on the initial presentation. We are grateful to the various researchers at Virginia Tech who loaned the USRPs used for fingerprinting.
This work was partly supported by NSF grants CNS-1410000 and CNS-1619728.

fingerprinting devices using features that vary due to changes in configuration. A proposed distance metric is developed which is invariant with respect to one or more configuration parameters. State-of-the-art features based on the frequency content of a signal are used, similar to those commonly used in the literature. When configuration is changed typical distance metrics, such as Mahalanobis distance, are unable to distinguish between devices. This is demonstrated on a collection of fifteen wireless transmitters using five modulation types, four bandwidths, and two carrier frequencies. It is shown that the proposed invariant distance metric allows re-identifying devices which have changed bandwidth or modulation. Using the invariant distances the median equal error rate decreases by less than three and in some cases even increases compared to the case when transmitters do not change configuration. The effects caused by changes in carrier frequency are also examined, and found to be minimal.

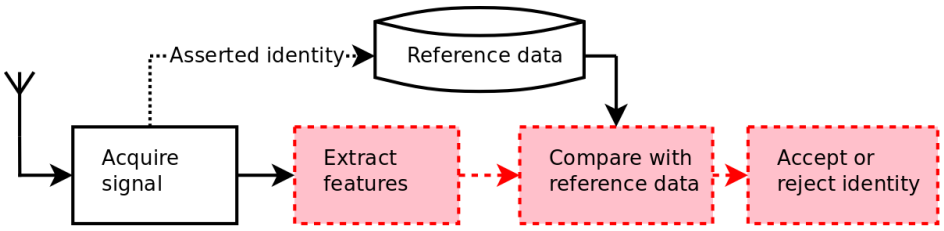


Figure 2.1: Steps in a fingerprinting setup: signal acquisition, feature extraction, comparison of the device under test with features known to have come from the asserted identity, and finally acceptance or rejection of the identity. Steps that may be impacted by device configuration are marked in red.

2.1 Introduction

Software defined radios (SDRs) are able to dynamically configure transmission parameters such as bandwidth, modulation, and carrier frequency. Cognitive radios (CRs) take advantage of this ability to dynamically choose the best configuration taking into account channel conditions, usage requirements, and other users [50]. As wireless transmitters find an ever increasing number of applications physical layer identification (PLI) methods have been proposed as an additional layer of security. PLI allows identifying devices based on small differences in each device's output that occur due to circuitry and manufacturing variation in their transmitter hardware. It is also referred to as device fingerprinting, in reference to traditional fingerprinting techniques that identify individuals based on unique biometric markers.

To verify an identity the PLI system, or fingerprinter, compares a device's signal with reference features known to have come from the asserted identity, as shown in Figure 2.1. However, when a CR alters their modulation type, bandwidth, or other configuration this may result in changes to the fingerprinting features used. When these features change substantially a comparison with the original reference data is not meaningful. We name this configuration dependency. Steps affected by configuration dependency are marked in red in Figure 2.1. Many features used for fingerprinting are negatively impacted by configuration dependency, including some features based on frequency content.

In this work we examine the challenge configuration dependency poses to PLI techniques in dynamic spectrum access (DSA) networks, and demonstrate a distance metric that is less affected by changes in transmitter configuration. We expand on the method initially presented in [49] (which only examined changes in bandwidth) and examine configuration dependency with regards to carrier frequency, modulation type, and bandwidth. Although this problem and our proposed solution is not specific to CRs, we consider it in the context of a DSA network using CRs as such a system may result in frequent changes to transmitter configuration. We describe the problem in greater detail in the next section, followed by the contributions of this paper and an overview of the paper's structure.

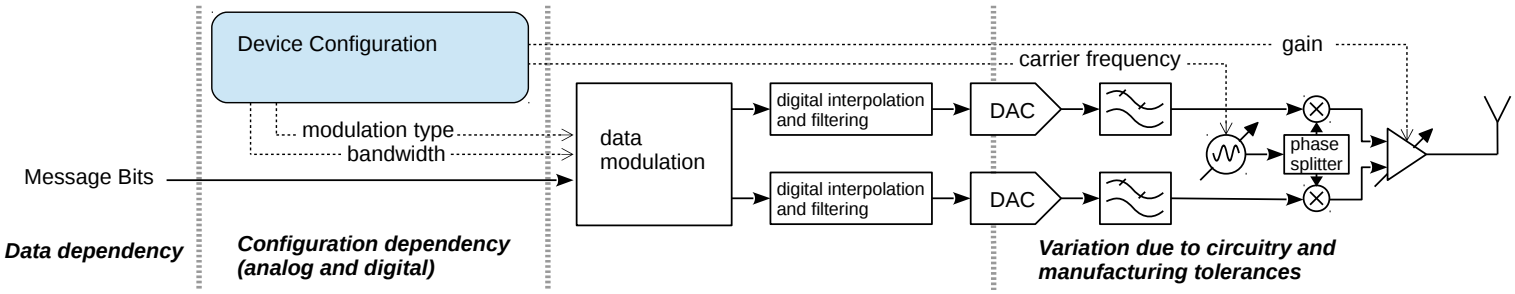


Figure 2.2: Simplified diagram of a transmitter showing sources of variation in fingerprints: data dependency occurrs when message bits cause variation; configuration dependency occurs when software configuration results in changing fingerprinting features; and the feature variation due to device hardware.

2.1.1 Configuration dependency

A simplified SDR transmitter is considered, shown in Figure 2.2. Effects introduced by the channel and fingerprinter's receiver are omitted, as well as some configuration of digital and analog components. The CR sends a message consisting of data bits. These bits are converted to IQ sample points using symbols from a given constellation. The digital sample points are converted to an analog signal and modulated with the carrier frequency to send a wireless signal. The final signal is a result of the message bits sent (called data dependency [2]); the baseband digital waveform created for a specific modulation type and bandwidth, and the parameters used in creating the analog signal (configuration dependency, including digital as well as analog components); and signal behavior introduced by manufacturing variation. Only the last contribution – from hardware variation – is desired for device fingerprinting. If the variation in a CR's fingerprint due to configuration dependency is much greater than the variation due to hardware differences current fingerprinting techniques will fail.

Fingerprinting features have been grouped in two types: predefined and inferred [2]. Predefined features are estimates of properties such as carrier frequency offset (CFO), modulation error, or IQ imbalance [45] that can be described, modeled, and measured. These features can be tied to the behavior of specific transmitter components including mixers and oscillators [10], the digital to analog converter (DAC)[11], and nonlinearity in the power amplifier [11, 51]. Since these features estimate underlying properties of the hardware, they are less

Expanded and revised from a paper published in Proceedings of the 2017 IEEE Conference on Communications and Network Security (CNS). Portions copyright 2017 IEEE.

This work expands on the method and results initially presented in [49]. We are grateful to the reviewers of this manuscript for their time and comments, as well as those who reviewed and commented on the initial presentation. We are grateful to the various researchers at Virginia Tech who loaned the USRPs used for fingerprinting.
This work was partly supported by NSF grants CNS-1410000 and CNS-1619728.

likely to exhibit data dependence. However, they have been shown to be more susceptible to feature replay attacks [36].

Inferred features rely on arbitrary transformations of the signal, such as the fast Fourier transform (FFT), the discrete wavelet transform (DWT), matched filter output [52], or inputting the signal to a neural network. Since these features generally rely on transforming the entire signal they are strongly influenced by changes in the signal, and are more likely to exhibit configuration and data dependency. Data dependency may be fixed by using a portion of the signal which has a constant bit sequence [2], such as the MAC address or synchronization symbols. Configuration dependency cannot be fixed in this way, as the entire signal sent may be changed. We have found that changes in transmitter configuration (primarily modulation and bandwidth) have a strong effect on inferred features based on the FFT, and may also affect other features.

2.1.2 Contributions

Our work, initially presented in [49],

- developed a method to compare fingerprinting features which is invariant to bandwidth
- demonstrated that state-of-the-art fingerprinting features based on the FFT fail to re-identify devices which have changed bandwidth
- demonstrated that the proposed invariant method by re-identifying CRs which have changed bandwidth, using data from 50 devices using eleven bandwidths collected over a wired channel

In this paper we expand on that work by

- providing a theoretical model for why the proposed metric provides invariant distances, and discussing when this metric is applicable
- discussing how to best choose the frequency features, and showing the method can handle different feature sets for each configuration
- demonstrating that the invariant distance can re-identify wireless devices which have changed configuration, including two different carrier frequencies, five modulation types, and a wider range of bandwidths

2.1.3 Paper structure

The remainder of the paper is arranged as follows: Section 2.2 describes a DSA network, and attacks that fingerprinting has been proposed to detect. We describe how current methods

will fail to detect these attacks when a CR changes configuration. In Section 2.3 preliminaries are covered including basic steps to perform PLI, several solutions to configuration dependency, and a model of how fingerprints are affected by changes in configuration. The proposed method to fingerprint CR devices with changing configuration is given in Section 2.4. Experimental results are demonstrated in Section 2.5. Related works are covered in Section 2.6, including the state of the art and several applications to CRs in DSA networks. Lastly, Section 2.7 contains conclusions.

2.2 System & threat models

We first describe a DSA network using CRs. Two attacks widely considered in the literature are described. The impact of configuration dependency on a fingerprinter's ability to detect these attacks is examined, and we outline the specific problems considered here.

2.2.1 Cognitive radio and dynamic spectrum access

One application of CRs is in DSA networks. Many portions of the radio spectrum are unregulated and overused, leading to interference among competing users. Other portions have been licensed to a "primary user (PU)", resulting in spectrum that goes unused at different times or geographical locations when the PU does not use it. DSA increases the overall amount of spectrum available by allowing opportunistic usage of licensed spectrum by secondary users. These users are allocated spectrum with certain restrictions including transmit time, power, or location. Users that violate these rules may have their allocations changed, or have their transmission jammed. This is to ensure that the PUs are not impacted by the system, and encourage secondary users to behave well [47].

Since users are allocated spectrum in part based on past behavior [46] it is necessary to identify and track transmitter's identities. Additionally, it is necessary for secondary users to be able to identify the PU to prevent interference with the PU's transmissions. This should be done without requiring changes to the PU. Due to these restrictions, PLI methods have been used in a number of ways to identify users in a DSA network [9, 53, 44]. However, these methods are unable to cope with changes in transmitter configuration at present. Several ways an attacker can exploit this shortcoming are discussed next.

2.2.2 Attacker capabilities

Two common attacks prevented by a PLI system are the Sybil attack and impersonation attack. We describe these attacks, and why changes in CR configuration can prevent these attacks from being identified. In the following PLI specifically refers to relating a known

device's identity to the device under test (DUT), while verification refers to determining if the DUT's asserted identity is correct.

Impersonation attack

An attacker has the ability to observe signals from other CRs, and is able to impersonate users at the digital layer by replaying observed bit sequences. When the spectrum has a licensed user, other users are must transmit on a non-interference basis, and vacate the spectrum when the PU transmits. An attacker impersonating the PU will prevent legitimate users from transmitting [54], and enjoy unrestricted access to the spectrum. This is the primary user emulation (PUE) attack. Device fingerprinting can be used to detect this attack [42, 9], and other impersonation attacks. Since the attacker has different hardware than the user they are impersonating, they will exhibit different behavior at the physical layer. This allows a PLI system to detect the attack when verifying the DUT's identity.

However, changes in configuration also cause different behavior at the physical layer. Fingerprinting systems with configuration dependency cannot distinguish between an attacker launching an impersonation attack and a legitimate device which has changed bandwidth or modulation type. If the possible configurations the legitimate device may use are known beforehand it may be possible to collect reference data for all configurations. In general, this is not the case.

Sybil attack

This attack occurs when a user assumes multiple identities to mislead the system. This could be done by a legitimate user who wants to avoid having some of their actions associated with them, or an attacker who has been identified previously and prevented from using the system.

Since an attacker's transmitter remains the same even as they assume new identities this attack can be detected using PLI techniques. However, if the features exhibit configuration dependency an attacker can assume multiple identities as long as each identity uses different configuration. Each configuration appears distinct to the fingerprinter. Consequently, a fingerprinter using features with configuration dependency will fail to detect this attack and be unable to tie attacks to a specific bad actor.

2.2.3 Problem statement

In the following we consider a PLI system using radio frequency features, or similar features with configuration dependency. The fingerprinter performs verification, preventing the impersonation attack. It can compare the DUT to all devices it has seen to find likely identities of an attacker, and preventing the Sybil attack from a known user. Due to configuration

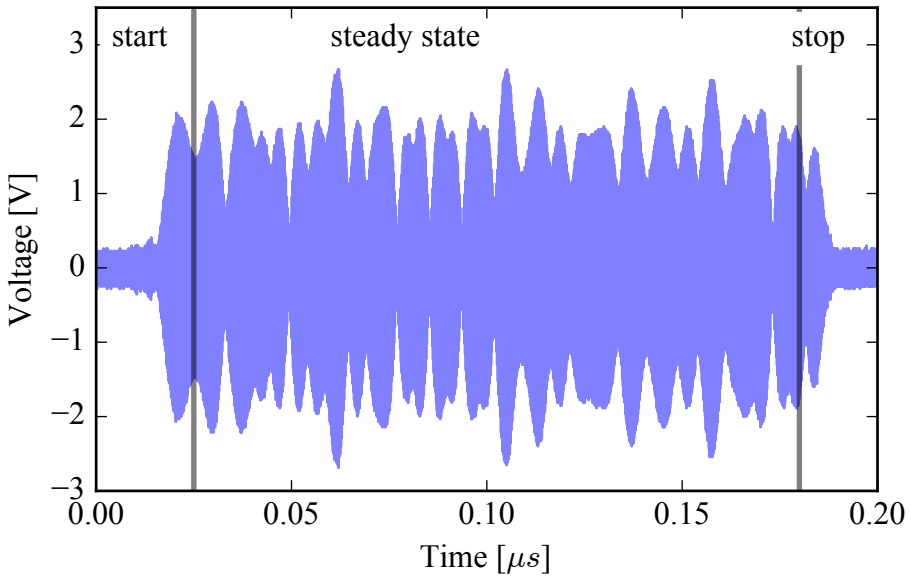


Figure 2.3: An example record, normalized to have zero mean and unit variance. The signal is 4QAM modulation at 0.25 MHz. The start, stop, and steady state portions are labeled.

dependency a legitimate user which changes configuration will no longer be recognized by the system. Similarly, the actions of an attacker which changes configuration cannot be tied to it. Consequently, we investigate methods to overcome configuration dependency.

2.3 Preliminaries

Before presenting an invariant distance metric we first establish some common notation, present the basic steps for device fingerprinting, cover several high level approaches to solving configuration dependency, and describe an approach to model configuration dependency which motivates our method.

A finite length discrete signal captured by an oscilloscope, such as the one shown in Figure 2.3, is called a record. The following notation is used:

- R^i a set of records from transmitter i
- R_P a set of records sent with configuration P
- L features extracted from a set of records, R
- $d(L^R, L^{DUT})$ distance metric between features from the reference (R^R) and test (R^{DUT}) records

τ threshold for validation

$\mathcal{K}$ a set of transmitters, $\{i, ..., j\}$

$D(L^{DUT}, L^{\mathcal{K}})$ indicates a function to find a vector of distances, $[d(L^{DUT}, L^1), ..., d(L^{DUT}, L^i)]^\top, 1, ..., i \in \mathcal{K}$

$|\bullet|$ the cardinality of a set

$l_{R,\bullet}$ the variation present due to the hardware of transmitter R in any configuration

$l_{R,T}$ feature value due to R's hardware only when operating in configuration T

$l_{\bullet,T}$ feature value due to an ideal transmitter in configuration T

2.3.1 Basic fingerprinting

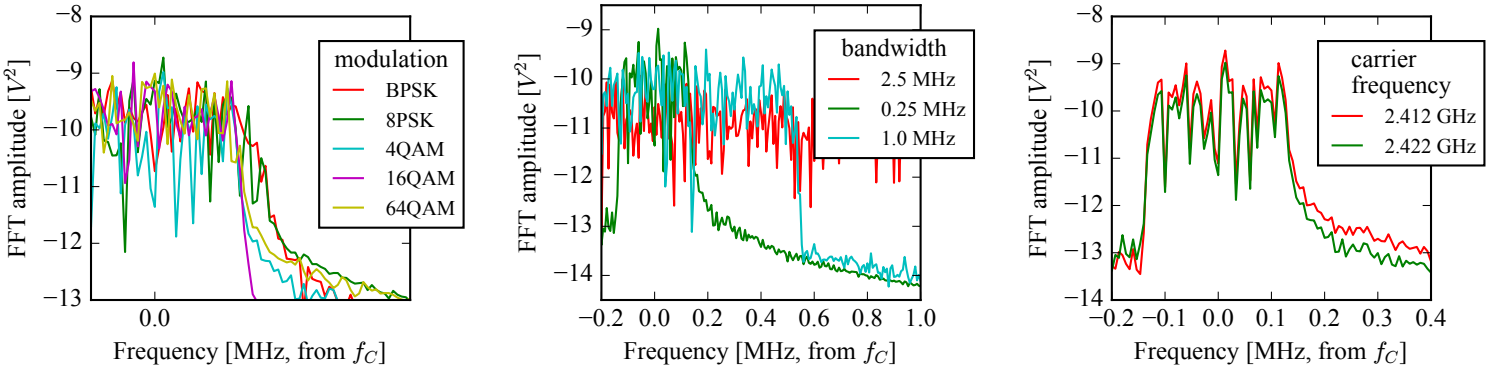
We first describe a simple approach to fingerprinting devices, similar to that used in other works [43, 9]. It is used with the proposed invariant distance metric, as well as existing distance metrics using state-of-the-art features. Before the fingerprinter can identify a device

1. users enroll reference data, L^R, in a database
2. a distance threshold, τ, is chosen based on the desired system performance

When a device (the DUT) asserts an identity the fingerprinter

1. extracts features from captured records, $R^{DUT} \rightarrow L^{DUT}$
2. looks up reference data for the asserted identity, L^R
3. calculates distance between the reference data and DUT, $d(L^R, L^{DUT})$
4. accepts the DUT's identity if it is similar to the reference data, $d(L^R, L^{DUT}) < \tau$, otherwise it detects an attack

When an attack is detected steps can be taken to identify the attacker or prevent the attack. The choice of τ presents a trade-off between the percentage of times an attacker is detected (based on the false accept rate (FAR)), and the impact of the system on a legitimate user (the false reject rate (FRR)). We define the FRR and FAR following [2]. The FRR is found as the percentage of time a valid user is incorrectly rejected for a given τ. Likewise, for a given τ, the FAR is the percentage of time an impostor's asserted identity is incorrectly accepted. What is an "optimal" value for τ depends on the system application - in some cases maintaining a low false accept rate is more important than a high true accept rate, or vice versa. When detecting the PUE attack τ would be chosen primarily based on the FRR so that there is a minimal impact to the PU's operation. This trade-off is usually expressed using receiver operating characteristics.



(a) (b) (c)

Figure 2.4: Variation caused by changing configuration in a single transmitter. Each line is the average of feature from 1500 observations. Configuration parameters varied are (a) Modulation: Small – but significant – changes are introduced in all features. (b) Bandwidth: Changes are largest in the sidelobes, but the bins in the mainlobe still have substantial variation. (c) Carrier frequency: The overall amplitude changes slightly.

2.3.2 Solutions to configuration dependency

When features exhibit data dependency the FAR or FRR will quickly increase beyond acceptable levels. Examples of variation due to changes in configuration is shown in Figure 2.4. Bandwidth (shown in Figure 2.4a) and modulation (Figure 2.4b) have the largest effect – causing far greater variation than that introduced by individual receiver hardware. The bins chosen are centered around the carrier frequency, and there is little variation due to the carrier as can be seen in Figure 2.4c. To verify the identity of the DUT when it uses a new configuration we consider several approaches:

i. Include examples of all possible configurations of a transmitter in the reference data.

ii. Use features that do not exhibit configuration dependency for the parameters changed.

iii. Develop a model, transformation, or distance metric which removes the affects of configuration.

Which approach is used will depend on what data is available, and how much configuration dependency features exhibit. Collecting reference data for each configuration (Approach i.) seems straightforward, and is used in [32]. All configurations must be known a priori: when the number of possible configurations is large this may require a substantial amount of time, storage, or other resources. Most importantly, this approach requires each device to willingly associate their identity with reference data for each configuration, making this an unsuitable method to identify attackers.

Using features without data dependency is appealing. A recent work [10] found changes in carrier frequency had a negligible effect on features including modulation error, and IQ offset. This makes Approach ii. plausible, at least for changes in analog configuration. However, we have found that larger changes due to bandwidth or modulation have a substantial effect on features, preventing identification or verification. We are unaware of any works that examine features when bandwidth or modulation is changed. Additionally, features that are not affected by configuration dependency may exhibit other drawbacks such as an inability to distinguish between many devices or lack of resistance to forgery.

Consequently, Approach iii. is investigated. This allows using common inferred feature types, such as the FFT, with transmitters which change configuration. The invariant distance metric can be used alongside naturally invariant features for greater distinguishability. We investigated several approaches including linear models and basing feature bin width on the DUT's bandwidth in [49]. However, these approaches failed to capture the complete interdependence between configuration and transmitter variation and are not considered further here.

2.3.3 Modeling configuration dependency

A simple model of transmitter variation is given to motivate various feature transformations. For ideal hardware components in an SDR the transmitter output can be modeled as a linear system which amplifies or attenuates the signal and has some frequency response. In practice, the effects of components can be nonlinear (e.g. power amplifiers [11]). Due to this difficulty we use a simpler model considering only the overall variation due to all components. Data dependency is not included, but could be treated similarly.

We consider transmitter R in configuration T. The estimated feature value can be modeled as the sum of three components,

$$L_T^R = l_{\bullet,T} + l_{R,T} + l_{R,\bullet} \tag{2.1}$$

where $l_{\bullet,T}$ is the feature value for an ideal transmitter in configuration T, $l_{R,T}$ is due to R's hardware only when operating in configuration T, and $l_{R,\bullet}$ is the variation present due to the hardware of transmitter R in any configuration.

For Approach ii. the features must not vary with configuration. The differences caused by changing configuration from S to T are smaller than the differences between transmitters R and DUT when

$$d(L_T^R, L_S^R) < d(L_T^R, L_S^{DUT}) \tag{2.2}$$

When this inequality is satisfied the features do not exhibit data dependency and devices which change configuration can be re-identified. The contribution of configuration T to a feature can be easily estimated by taking the mean feature value for a specific configuration

over all devices. Thus, l_T can be compensated for. By letting the distance metric be translationally invariant (such as the norm) parts common to transmitter R can be removed from the distances. The resulting relation is then

$$d(l_{R,T}, l_{R,S}) < d(l_{R,S} + l_{R,\bullet}, l_{DUT,S} + l_{DUT,\bullet}) \tag{2.3}$$

This inequality does not hold for features based on the FFT: For changes in bandwidth we have experimentally verified that the variation between configurations for a single transmitter ($d(l_{R,T}, l_{R,S})$) is greater than or equal to the variation between transmitters using the same configuration ($d(l_{R,S} + l_{R,\bullet}, l_{DUT,S} + l_{DUT,\bullet})$). When configuration is changed verification cannot be performed with frequency features. Instead, a two-step distance metric which is invariant to these changes is developed.

2.3.4 Hypothesis of invariance

Predicting the quantities needed to model Eq. 2.1 is difficult. Instead, we develop a distance metric which can be invariant to changes in configuration, based on the hypothesis that transmitters should have consistent behavior as configuration changes. This requires reference data from a small number of devices in each configuration considered, but does not require knowing $l_{R,S}$ or $l_{DUT,S}$.

Fingerprinting features are due to variation in the hardware of each device. The hardware should not change behavior as configuration changes. The settling time of the DAC is much less than the symbol period, suggesting that even as the digital waveform changes there will be little change in the DAC's behavior seen by the fingerprinter. The amplifier and filter responses are also unlikely to change substantially over the bandwidths and carrier frequencies considered. Consequently, the relationships between two transmitters using the same configuration should be fairly consistent, regardless of which configuration is used. Specifically, if $d(L_T^R, L_T^{DUT}) >> 0$ then $d(L_S^R, L_S^{DUT}) >> 0$. Similarly if $d(L_T^R, L_T^{DUT}) \approx 0$ then $d(L_S^R, L_S^{DUT}) \approx 0$. That is, devices which are very similar under configuration S will continue to be similar under configuration T. Likewise, devices which are very different at T will be very different at S.

This hypothesis can be examined experimentally in several ways. If we consider the features extracted from records created by CRs using a single configuration to be in a subspace of the larger space of features for all possible transmitter configurations, then principal component analysis can be used to examine these subspaces. Features which are zero-mean for each configuration are used to prevent each configuration's power level (l_T and l_S) from dominating the analysis. This is shown in Figure 2.5 for changes in bandwidth. The components do not correspond directly the model in Eq. 2.1. But, it can be seen that bandwidth has the largest effect on the features, while variation due only to each transmitter (l_R) contributes relatively little. Relations between transmitters remain consistent even though useful comparisons

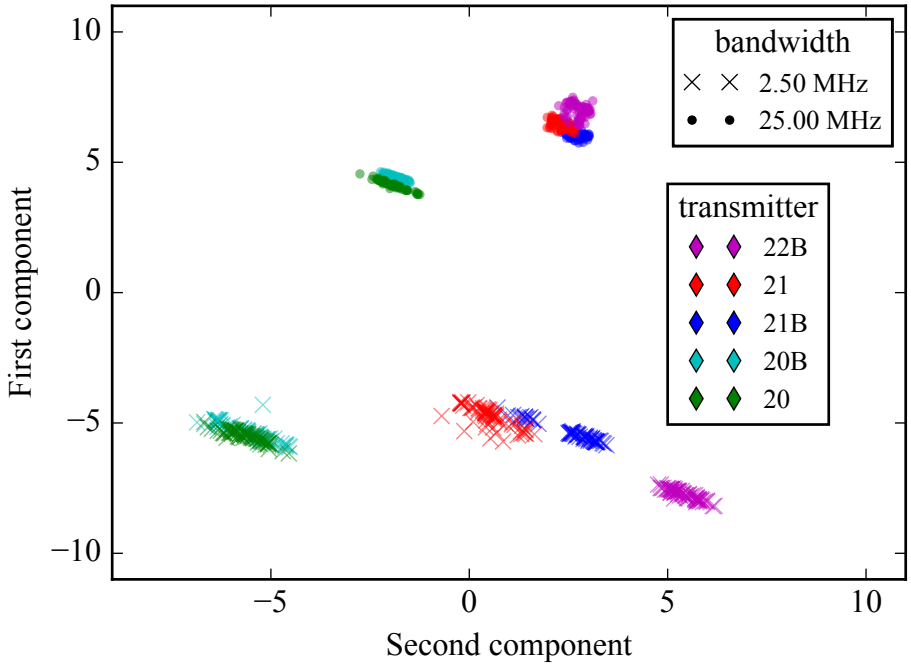


Figure 2.5: Top two principal components for zero-mean features from SDRs transmitting a 4QAM signal at two bandwidths. The effects of bandwidth and individual transmitter characteristics are visible, while relative distances between devices are similar at either bandwidth.

with different configuration are not meaningful. This confirms our previous hypothesis, and suggests that the invariant distance metric will provide good results.

2.4 Invariant distances

We describe an approach to calculating distances which are invariant to changes in CR configuration, determine the amount of reference data it requires to re-identify devices, and analyze the impact on security.

2.4.1 Developing an invariant distance metric

Given a DUT operating in a configuration without reference data we name this configuration the target, T. A fixed set of transmitters, $\mathcal{K}$, with data enrolled using the target configuration is found. The distance of the DUT to each device in the fixed set is found for the target configuration, denoted $D_T = D(L^{DUT}, L_T^{\mathcal{K}})$. A source configuration is chosen which has data enrolled for the fixed devices and the asserted identity. The distance is measured to the same fixed devices and the asserted identity of the DUT with the source configuration, $D_S = D(L^R, L_S^{\mathcal{K}})$. The resulting distances to devices in $\mathcal{K}$ for the target and source configuration are compared, $d(D_S, D_T) < \tau$.

We have found that Euclidean distance works well to find D_S and D_T. Figure 2.6 shows distances to nine transmitters at two bandwidths using Euclidean distance. Since these distances vary in magnitude at the source and target bandwidths cosine distance works well for comparing D_S and D_T as it ignores magnitude and instead focuses on differences between each feature.

In summary, the identity of the device under test can be verified as follows:

1. The target configuration, T, corresponds to the configuration of the DUT
2. A configuration where the asserted identity has enrolled data is the source, S
3. Fixed transmitters are chosen which have enrolled data at T and S, $\mathcal{K}$
4. Distances from each fixed transmitter to the DUT are found, $D_T = D(L^{DUT}, L_T^{\mathcal{K}})$
5. Distances from each fixed transmitter to the asserted identity are found, $D_S = D(L^R, L_S^{\mathcal{K}})$
6. Use $d(D_S, D_T)$ with appropriate threshold, τ, to verify the DUT's identity

The distances, D_T and D_S, can then be compared using a second distance metric. These steps form a distance metric which is less impacted by changes in transmitter configuration. Since features are only compared directly when they come from signals created using

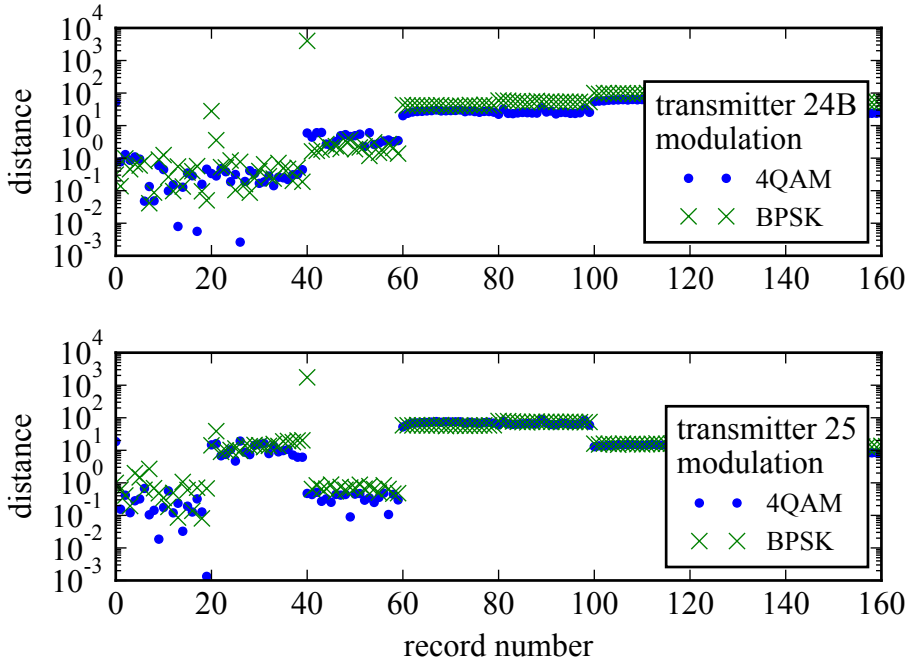


Figure 2.6: Distances to fixed transmitters, for two different modulation types. Two transmitters shown, with 20 records per fixed transmitter. Similar behavior can be seen in other transmitters.

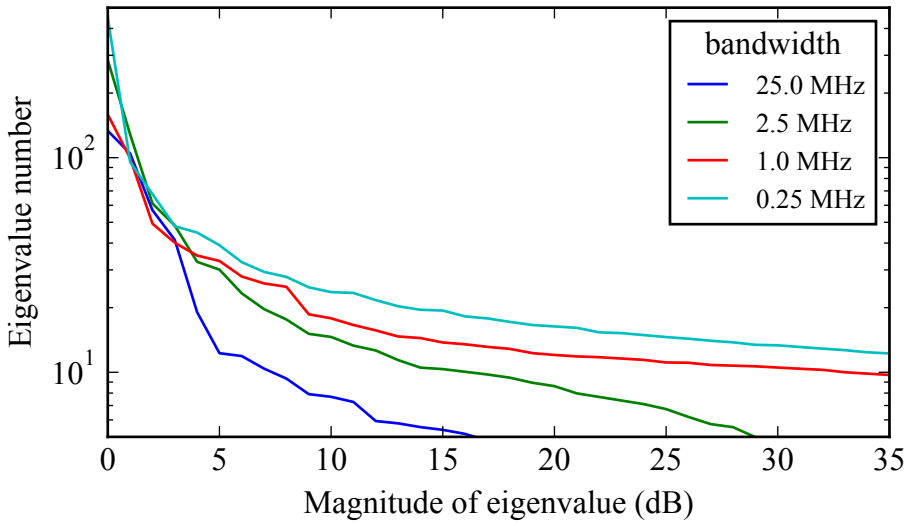


Figure 2.7: Sorted eigenvalues of the zero-mean features at each bandwidth, using features from all devices. Most of the variance in the data is concentrated in the first dimension, and decreases by an order of magnitude in the first 5 to 10.

the same configuration this also opens up the possibility of using different feature sets for each configuration. This allows the invariant distance metric to compare source and target configurations using different feature sets.

2.4.2 Choice of $\mathcal{K}$

The number of fixed transmitters used, $|\mathcal{K}|$, will have a substantial impact on processing complexity and the amount of enrolled data required. Before presenting experimental results on the number of fixed transmitters we find it useful to estimate this number based on the features used. The eigenvalues of a set of features reflect the amount of variance contained in each dimension of the data. Larger eigenvalues indicate more variation in the data, while smaller values typically correspond with noise. The eigenvalues of the features at a single bandwidth will describe the dimensionality of the subspace at that bandwidth. The eigenvalues were examined using zero-mean features from all transmitters at four bandwidths, shown in Figure 2.7. It was found that the first dimension contains the most variation, and the majority of variation in the data is contained in the first five to ten dimensions. Consequently, it should be possible to re-identify devices using five to ten fixed transmitters.

The choice of devices included in $\mathcal{K}$ may also have an impact on performance. A device might provide features that result in inconsistent distances to other transmitters, leading to poor performance with the invariant distance metric. Methods of choosing the devices in $\mathcal{K}$ were examined in [49], where it was found that most devices work equally well. Consequently,

devices were randomly chosen for inclusion in $\mathcal{K}$ in this work.

2.4.3 Analysis of mutual information

Mutual information provides a measure of how much information each feature provides to distinguish between transmitters. The mutual information does not relate directly to fingerprinter performance – actual performance will depend on the number of features used and the distance metrics used. However, the mutual information should provide a theoretical bound on performance for a given set of features, so it is used to examine the effects of the invariant distance metric.

Mutual information is a non-negative number, and found between features and transmitter identity using binned feature values as

$$MI(L, C) = \sum_{i=1}^{N} \sum_{j=1}^{M} P(L_i, C_j) \log \frac{P(L_i, C_j)}{P(L_i)P(C_j)} \tag{2.4}$$

Where C is the identity of transmitters in a binned feature L, and there are N observations belonging from one of M transmitters. The choice of bins can have an effect on this calculation, and so the same set of bins is used for all bandwidths with each feature. The resulting calculations are shown in Table 2.1.

The original features provide a similar amount of information at each bandwidth, and lowers somewhat when features from all bandwidths are used. When bandwidth changes the fingerprinter's original distances have a decrease in mutual information. This suggests that even a perfect fingerprinting system will experience a decrease in performance when devices change bandwidth. If this were not the case fingerprinter could re-identify the DUT after a change in bandwidth it would be expected the mutual information would remain the same. The intermediate features used by the invariant distance metric provide an increase in information, except when features from all bandwidths are included. However, they provided similar performance to the original features. This is probably due to the smaller number of intermediate features providing less discriminatory power overall. Performance using these features across all bandwidths is much better than the original distances, which is not suggested by the mutual information values. This suggests that the performance of cosine distance used with the invariant distance metric is not described well by mutual information.

2.4.4 Impact on security

An attacker that observes the DUT can attempt to copy its characteristics. This includes recording and replaying IQ data in a signal replay attack, or recording signal characteristics and attempting to match them in a feature replay attack [36, 37]. An attacker can

Table 2.1: The mean mutual information (and standard deviation) of all features for several bandwidths. The original features denotes mutual information between features and transmitter identity using the raw RF features, for a DUT at the same bandwidth as the reference data. The original distances are the fingerprinter output for Euclidean distance when the DUT's bandwidth changes from that used in the reference data, and can be seen to decrease substantially from the original features. Lastly, the mutual information between transmitter identity and the intermediate features used by the proposed invariant distance metric are given. This increases the mutual information above the original features.

bandwidth	original features	original distances	invariant distances
2.5 MHz	1.77 ± 0.30	1.22 ± 0.03	2.19 ± 0.11
1.0 MHz	1.69 ± 0.26	1.15 ± 0.02	2.12 ± 0.14
0.25 MHz	1.90 ± 0.15	1.32 ± 0.11	1.94 ± 0.11
all	1.62 ± 0.14	—	1.70 ± 0.13

also attempt to obtain hardware that behaves similarly to the DUT [2]. This is a greater vulnerability when consumer grade hardware is used.

The proposed method is no more vulnerable to these attacks than existing methods. Steps (3) and (5) of the method find distances between devices using existing features. Consequently, an attacker with the ability to fool an existing fingerprinting system would be able to fool our system in these steps. However, predefined features based on modulation have been shown to be most vulnerable to these attacks [36, 2]. Our method increases overall security by allowing the use of frequency based features, which are more resistant to these attacks. These features could not be used otherwise due to data dependency.

It may be possible that an attacker can devise an input that the invariant distance metric fails to detect, that other distance metrics would successfully detect. In the worst case, the invariant distances are vulnerable to this when the number of fixed transmitters is less than the number of features used. For most feature types this will be the case. However, following the analysis of the previous section that five to ten fixed transmitters can reliably identify an attacker using hardware similar to the DUT for a feature replay attack. In general, an attacker exploiting this weakness would require some knowledge of how the fingerprints are created and what information is discarded in dimensionality reduction. This would require considerably more effort than a feature replay or signal replay attack. This is a general weakness of any method which reduces the dimensionality of fingerprinting feature: by reducing the features information that allows identifying an attacker could be inadvertently discarded.

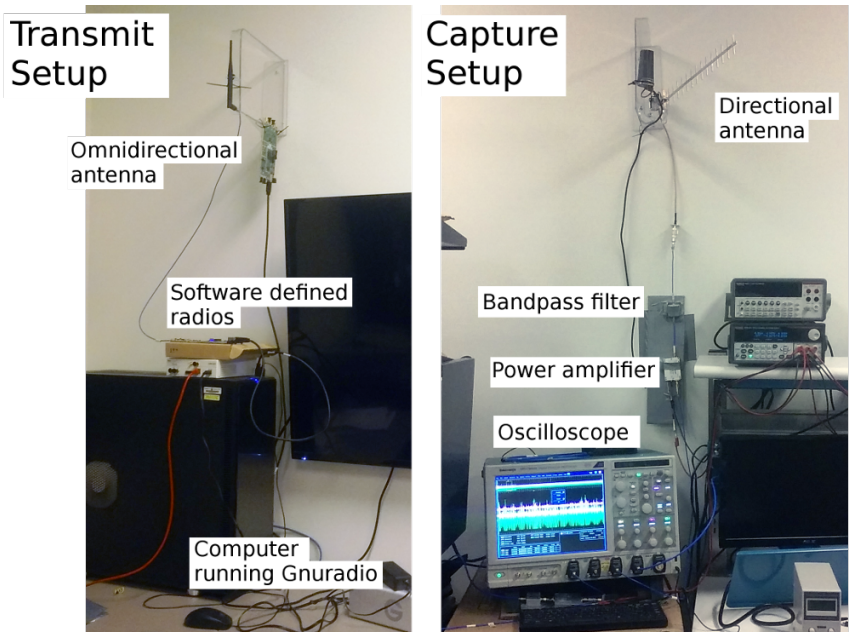


Figure 2.8: Experimental setup used. Transmit (omnidirectional) and receive (directional) antennas are opposite each other in the same room. Figure taken from [1], which re-used this experimental setup.

2.5 Experimental evaluation

We present results of running the method on wireless data collected in an indoor environment. Data is collected from fifteen SDR devices, switching configuration between four bandwidths, using five modulation types and two carrier frequencies. First, we describe how this data is collected. This is followed by a description of the radio frequency features, and details of feature selection. The evaluation of the RF features using Mahalanobis distance, and these features used with the invariant distance metric is described and performance given. We find that when modulation or bandwidth vary the performance of the invariant metric is slightly below the original features. When carrier frequency varies performance can exceed the original features. Lastly, comparisons with data used in [49] is given, and the impact of the choice $\mathcal{K}$ is discussed.

2.5.1 Signal capture setup

The setup, shown in Figure 2.8, consists of a transmit and receiving antenna, a computer running GNU Radio used to control the SDRs, and an oscilloscope to capture records. QAM and PSK modulation are used with bandwidths between 250 kHz and 10 MHz with constellation sizes from 2 to 64, as described in Table 2.2. The digital signals are created in GNU Radio version 3.7.10.1 using the QAM Mod and PSK Mod blocks on a computer

running Ubuntu 14.04 LTS.

The data sent is a constant sequence of bits which are randomly generated beforehand and re-used for all transmissions. The data is sent with a sufficient pause between subsequent transmissions to allow the oscilloscope to capture, check, and save the signals of interest.

To minimize the effects of the channel stationary transmit and receive antennas are used. Each DUT is attached to the transmit antenna by an SMA cable, with gain chosen so that all transmitters have similar power. The receiving antenna is a directional antenna located at a distance of 4.6 m from the transmitting antenna. The signal from the receiving antenna is fed through a bandpass filter covering the ISM band, low noise power amplifier, and attenuator. This brings the signal into a range the oscilloscope can sample, and removes some unwanted signals present in the wireless environment.

The oscilloscope used is a Tektronix DPO7354C [55], running Matlab scripts to capture data and invoke GNU Radio with the configurations tested for each transmitter. A sampling rate of 25 Gs/ sec is used, while the triggering level was adjusted as needed for each transmitter. Fifteen transmitters are used, shown in Table 2.3. The N210s [56] were used with daughter-cards having a single transmit frontend. The B210s [57] have two transmit frontends, which are considered as separate devices in the following experiments.

Records of length 5×10^6 samples were captured, consisting of noise before transmission, the signal transient, the steady state portion, and noise after the signal of interest, shown earlier in Figure 2.3. Since the signals were collected from a wireless environment with other wireless signals present several precautions were taken to ensure each record captured includes the signal of interest: A relatively high triggering level was used, the signal was bandpass filtered to the ISM band, and several checks were applied to each captured record. The checks used were refined between some data runs to improve performance, but all are based on power thresholds. Records which had a high power level in the noise before or after the signal were discarded, as well as records that had too low of a power level over the steady state portion or did not exhibit a change in power level where the signal start should have been. This prevented the majority of spurious data being saved. By running the capture setup when the signal of interest was not present it was found that on average 1 in 1000 records was accepted. This implies that less than 1 % of records are spurious, provided the wireless environment did not change substantially between data runs (which seems to have been the case). A total of 1500 records were gathered from each transmitter in each configuration.

2.5.2 Frequency features

The features used are extracted from the frequency content of the steady state portion of the signal. Features based on frequency content provide good performance, and have been used in a number of works including [2, 43]. We describe them in detail here, as the specifics of frequency features vary in the literature.

Table 2.2: Configurations tested. A total of four bandwidths, five modulation types, and two carrier frequencies are used, for a total of fifteen data runs.

Modulation	Bandwidths (MHz)	Carrier(GHz)	Total
4QAM	10, 2.5, 1, 0.25	2.422	4
4QAM	2.5, 1, 0.25	2.412	3
16QAM	0.25	2.422	1
64QAM	2.5, 0.25	2.412	2
BPSK	2.5, 0.25	2.422	2
8PSK	2.5, 0.25	2.422	2
total			15

Table 2.3: Transmitters used for fingerprinting. Some models of USRP provide multiple transmit frontends, which are counted separately.

Model	Number of SDRs	Daughtercard or frontend	Number of datasets
N210r4	3	SBXv3	2
		UBXv1	1
B210	23	A	12
		B	12
total	29		15

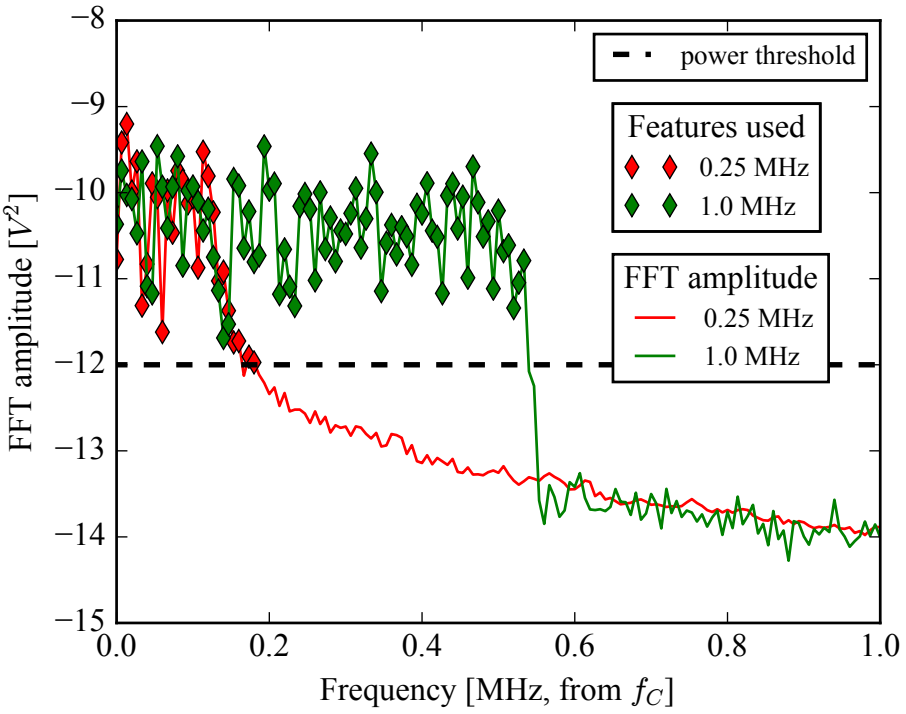


Figure 2.9: Illustration of how features are selected using average bin power. FFT bins with amplitude above a threshold are used as features, while other bins are discarded. This causes different features to be used for each bandwidth.

Each record consists of a sequence of voltage codes. The transient portion of each record is discarded, leaving the steady state portion (beginning 2×10^3 sample points after the detected start of the signal). From this 4×10^6 points (160 µs) are taken, which are normalized to have zero mean and unit variance. The FFT of these samples is taken, and bins covering 2 MHz centered at the carrier frequency are selected. The logarithm of the magnitude of each bin is taken as features. This results in 300 bins each with a width of 6.67 MHz.

Variation in the mainlobe of the frequency spectrum seems to provide the best features for fingerprinting, while the sidelobes provide less effective features due to attenuation. The noise present outside the mainlobe of the data also has a substantial negative impact on performance. Consequently features used are limited by a threshold on signal power, shown in Figure 2.9. Features with average power below the threshold are discarded. The threshold is chosen to be larger than the noise floor, and (somewhat arbitrarily) about 4dB less than the highest bin power. This improves the overall performance. At narrower bandwidths, this results in a smaller set of features. The set of features used as modulation and carrier frequency vary is constant.

2.5.3 Evaluating the original features

Mahalanobis distance has been used successfully in other fingerprinting works [43]. It is used here to provide performance when using the original features, and compare with the state of the art. The reference data consists of 800 records for each device and configuration. Each experiment was run ten times, with a randomly selected (but continuous) subset of records as the reference set.

To summarize the performance of these runs for all transmitters results in this and all following sections are presented as the cumulative distribution of equal error rate (EER). The EER ranges from 0 to 100, and is determined by finding the threshold, τ, for which the FAR and FRR are equal. It is found for each DUT using a target configuration, versus all other devices using the same configuration. In a system with good performance it should be near zero, meaning that both the FAR and FRR are low. Generally the EER is unsuitable for directly comparing classifiers [58, Chapter 5], but it is an easy way to summarize the performance of multiple systems. The cumulative distribution function expresses what percentage of EERs fall below a given rate.

Figure 2.10 shows the cumulative EER for the original features. It is expected that this will be an upper bound on performance with changing configuration. Also important is the variance of the EER. To determine the variation of the EER we compare the 25th and 75th percentiles of the EER to the median. This provides a measurement of variation, similar to the standard deviation, but is more suitable for non-normal data (one standard deviation encompasses 68 % of the data in a normal distribution, while the percentiles used encompass 50 %). In the following, when variation is given without specifying the percentiles it refers to the 25th and 75th percentile. The median EER is around 2, with variation of 1 and 4, a difference of 1 and 2 from the median EER. This shows that there is very small variation in the EER, and performance will be good across all runs and sets of reference data. This corresponds to the higher carrier frequency. At a lower carrier frequency the median increases to 7, probably due to the antenna's frequency response. The variation remains similar, with a difference of 1 and 2.5 from the median.

2.5.4 Evaluating the invariant distance metric

We evaluate the method using the frequency fingerprints described previously. The invariant distance metric uses a combination of two other distance metrics: The distance to each reference transmitter is calculated using Euclidean distance with reference data selected as described in Section 2.5.3. Cosine distance is then used to compare these sets of distances. This compensates for a difference in amplitude between configurations, although the relationships between transmitters are approximately constant.

The source and target configuration for each test is chosen by taking all pairwise combinations of the configuration parameter being varied. Each device is taken as the DUT for each

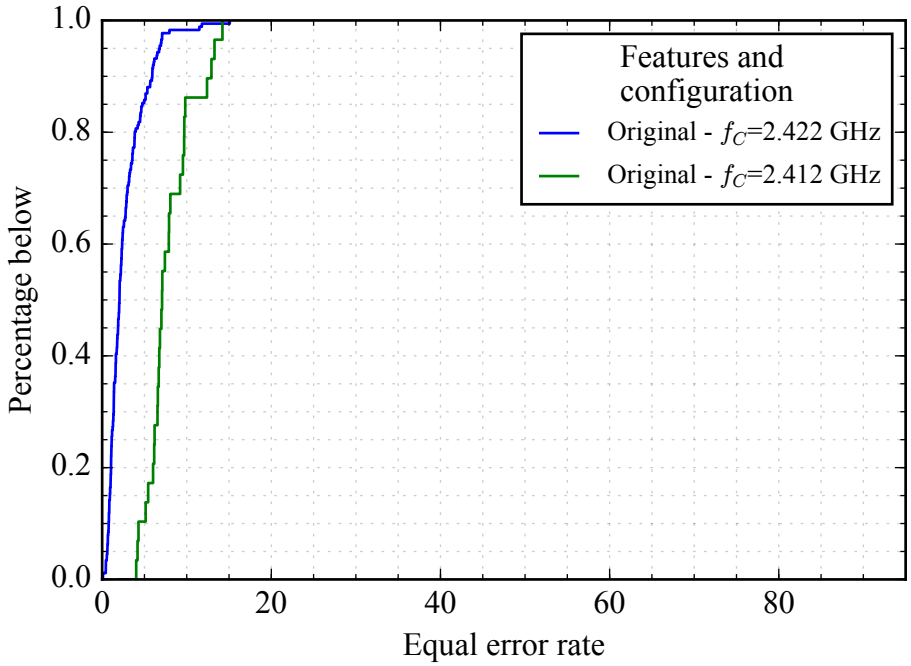


Figure 2.10: Performance using state-of-the-art frequency features when configuration does not change. Carrier frequency impacts performance, and so is separated by parameter. This is not the case for bandwidth and modulation: the results reflect all considered configurations of these parameters.

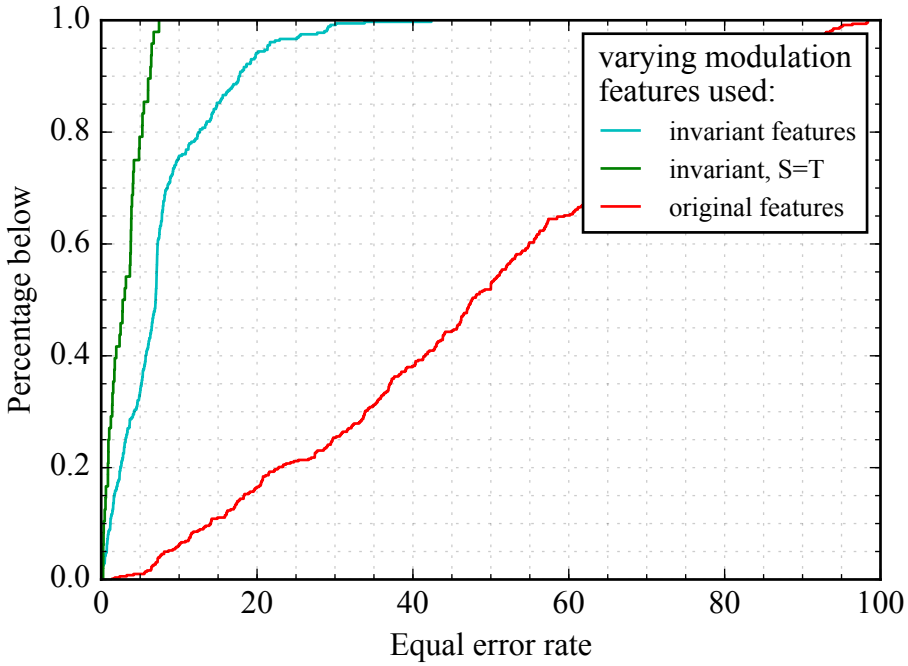


Figure 2.11: Performance when modulation type changes. The invariant distance metric allows for verification, while the state of the art (original features) fails when modulation type changes.

target configuration, and the EER is found for each possible source configuration. Only a single configuration parameter is varied at a time. Configuration parameters varied are modulation, bandwidth, and carrier frequency, shown in Table 2.2. For example, when the target modulation is 16QAM there are four possible source modulation types with a configuration using the same bandwidth and carrier frequency. Except where otherwise specified, ten devices are used as fixed transmitters. The fixed transmitters are chosen randomly but do not include the DUT.

2.5.5 Performance

Next, we consider the performance when transmitters change their configuration. Performance is considered for three cases: using the original radio frequency fingerprints directly when transmitter configuration is changed; using the invariant distance metric transmitter configuration is changed; and performance of the invariant distance metric when configuration is not changed (although it is not needed in this case using it may reduce the amount of reference data a fingerprinting system must store). Performance for configuration of modulation type, bandwidth, and carrier frequency are shown in Figures 2.11, 2.12, and 2.13 respectively.

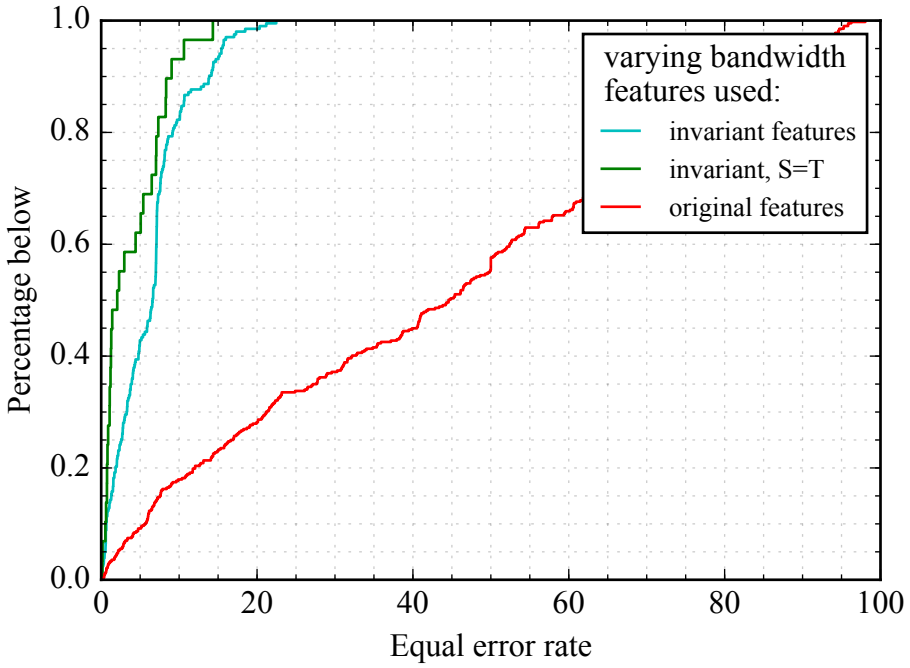


Figure 2.12: Performance when bandwidth changes. The invariant distance metric is slightly better than when modulation type changes. The original features cannot distinguish between devices.

Changes in bandwidth and modulation affect performance very similarly. The original frequency based features are completely unable to handle changes in bandwidth or modulation. They exhibit a median EER around 45. This is effectively random behavior, which is also reflected by variation of 23 and nearly30 from the median. In a practical system this means it would be impossible to verify the DUT's identity. The invariant distance metric allows verifying identities after changes in transmitter configuration with performance only slightly worse, on average, than that given by the original features for a single configuration. For changes in modulation the median EER is under 7, with variation of 4 and 0.5. This is somewhat higher than the median EER using Mahalanobis distance, but the variation is comparable. For changes in bandwidth the median is also under 7, with variation of 4 and 1. This is somewhat surprising, as due to the power based threshold for feature selection different feature sets are used for each bandwidth, while the same feature set is used for each source and target modulation. It would be expected that different feature sets would lead to worse performance, however this shows that the proposed method can have good performance even when the source and target data use different sets of features.

When configuration is not changed but the invariant distance metric is used performance is better than when configuration changes, but slightly worse than the original features without changes in configuration. The median EER is around 2 to 3. If necessary, it would be possible to use only the invariant metric without a large loss in performance. This would allow a fingerprinting system to store features from fewer devices in each configuration.

Changes to the carrier frequency give more interesting results. When carrier frequency is changed using the original frequency features directly increases the median EER by less than 3, although the difference to the 25th and 75th percentile increases to 4 and 3. This shows that the variation between runs increases, as well as the average EER.

Performance is improved when using the invariant distance metric. The median EER decreases by over 7 for f_c =2.422 GHz, while it increases slightly (by about 4) at 2.412 GHz. There is not a large change in variation. This shows that the invariant metric can improve the performance of the original features in some instances, which was not expected. Using the invariant distance metric without varying configuration also has a higher EER than the original features.

Overall the results show that our method makes it possible to verify device identities with a high degree of accuracy even when device configuration changes. The original frequency features, used in state-of-the-art works, exhibit configuration dependency and cannot distinguish between devices. Using these features with the proposed invariant distance metric, the fingerprinter can verify the identity of devices that change configuration with nearly the same accuracy as if they had not changed bandwidth, modulation, or carrier. This confirms the hypothesis in Section 2.4 that device behavior at the physical layer remains consistent even as configuration changes.

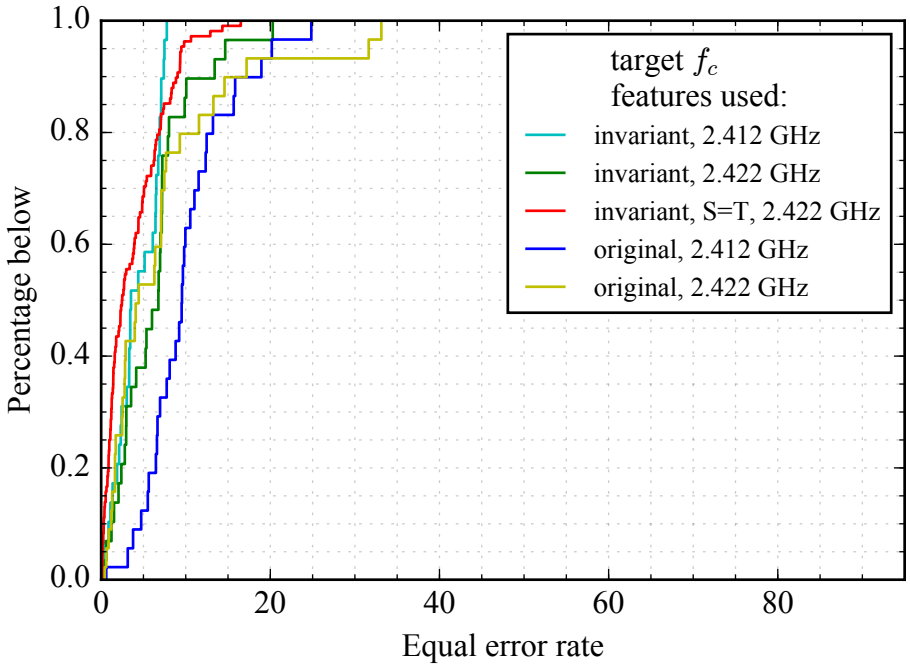


Figure 2.13: Performance when carrier frequency changes, split by target configuration due to difference in performance. Although the original features perform quite well, the invariant metric provides a slight increase in performance for both target frequencies.

2.5.6 Comparison with wired data

We provide a brief comparison with the preliminary results in [49], which used data collected over a wired channel. The original features without transmitter configuration changing have slightly better performance in the wired case. This is expected, as there is less noise in the environment. Surprisingly, the performance using the invariant metric and varying bandwidth was slightly worse when using the wired channel. The median EER was over 15, with 10 % over 22 - over 7 higher than the wireless data for both points. The better wireless performance is probably due to the additional feature selection step in Section 2.5.2, which was introduced since noise outside the mainlobe of the frequency spectrum had a substantial negative impact on performance. It could also be due to the more limited amount of reference data collected over the wired channel.

2.5.7 Selecting devices in $\mathcal{K}$

We examine the choice of fixed transmitters in R. Performance as the size of $\mathcal{K}$ varies is shown in Figure 2.14. It can be seen that as the number of transmitters increases performance improves with diminishing returns. This supports the hypothesis advanced in Section 2.4.2 that five to ten fixed transmitters should give good performance.

As demonstrated in the previous section, the invariant distance metric gives acceptable performance even when configuration is not changed. Consequently, we consider the minimum amount of reference data the fingerprinting system must maintain. Consider a system with M devices and P possible transmitter configurations. If features without data dependency can be used (Approach ii.), only M sets of reference data (one per device) are required. Using the proposed method $M+(P-1)*|\mathcal{K}|$ reference sets are required. This is linear in either the number of configurations, or number of devices. Provided the set of fixed transmitters, $\mathcal{K}$, is small a large number of devices and combinations can be efficiently handled. Using standard fingerprinting techniques with each device enrolling reference data for every possible configuration (Approach i.) would require MP reference datasets to accurately identify each device. The amount of reference data required increases combinatorially with the number of devices and possible configurations.

Our method requires more reference data than features not exhibiting data dependency, but considerably less than that required to have reference data for every possible configuration. Although the method reduces the amount of data needed, most importantly it allows identifying users that change configuration, such as would occur in an impersonation attack.

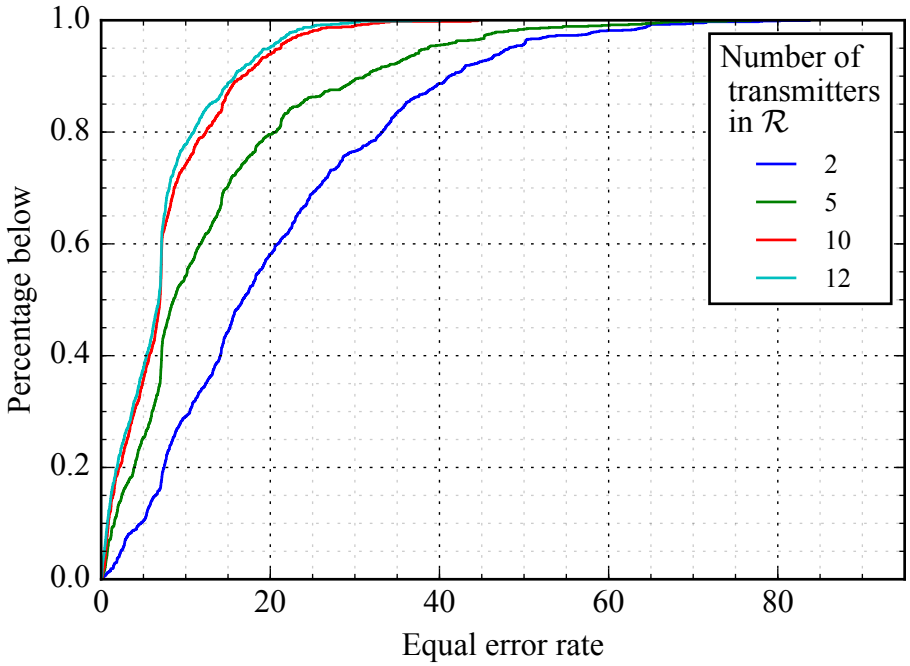


Figure 2.14: Cumulative distribution of EER for various sizes of $\mathcal{K}$. Note the diminishing returns as the number of transmitters used for reference distances increase.

2.5.8 Extending to other types of features

We evaluate under what conditions features are likely to work RF features used have been shown to incorporate CFO in some instances, including for the devices fingerprinted here. Consequently, examine how this may be contributing to the invariant features and try to generalize to other types of devices.

MODEL here

2.6 Related work

We cover related literature, primarily on device fingerprinting at the physical layer. The state of the art for physical layer identification has been covered somewhat recently in [2]. Fingerprinting methods specific to wireless devices are covered in [3], including methods using characteristics of higher-level protocol layers such as the MAC layer. Approaches not based on fingerprinting have been proposed that use the physical layer to secure DSA networks. Information related to traditional user identities can be embedded at the physical layer by slightly modifying each transmitter's signal in a manner that is transparent to receivers. Properties modified include the constellation [59] and frequency offset [28]. These

approaches have abilities and drawbacks similar to traditional identifiers. It is also possible to use properties of the channel to identify users, such as received signal strength (RSS) [8], angle of arrival [60], or link signatures [61]. Unlike device fingerprints these properties are not specific to a single device, consequently an attacker cannot be distinguished from a legitimate user located nearby.

We only cover fingerprinting methods at the physical layer, as these are most relevant to this work. These include features and classifiers currently in use, as well as types of devices that have been fingerprinted. Recent works proposing theoretical limits, and modeling sources of variation are explored. Finally, recent applications specific to DSA networks are given.

2.6.1 Current fingerprinting methods

Features that have been used to identify transmitters include instantaneous frequency, clock skew, transient length, timing errors, and wavelet coefficients [2]. These can be extracted from the steady state or transient portion of a record. The Fourier transform and wavelet decomposition both allow easily extracting a large number of features from a signal, and can provide high accuracy when distinguishing between devices. A diverse number of wireless devices have been fingerprinted [2], including Bluetooth, GSM, VHF, UHF, and WiFi transmitters. Error rates (calculated as percentage of incorrectly classified records) vary from less than 1 % to nearly 30 %, generally with devices of the same model [2].

Only a few works have examined transmitters changing configuration. Fingerprinting transmitters using multiple carrier frequencies is examined in [10], as previously mentioned. In [32] a convolutional neural network is used to learn features to identify transmitters at multiple bandwidths simultaneously. The features learned provide a high degree of accuracy for classification, but are dominated by bandwidth when multiple bandwidths are used. This requires the fingerprinter to be trained with data from each bandwidth for every transmitter (Approach i.). Fingerprinting with multiple modulation types is examined in [62], however the modulation type is unique to each transmitter. This simplifies identification, as each modulation type introduces additional variation in the features. Although not related to configuration, in [16] wireless access points are identified using clock skew. This variation originates in each device's oscillator and is heavily impacted by temperature. A method to compensate for temperature based change by measuring between multiple devices is given.

2.6.2 Theory of fingerprints

Applying PLI in systems with an increasing number and diversity of devices will benefit from theoretical guarantees on performance. Likewise, modeling the causes of device variation will help determine how resistant features are to forgery, and how feasible it may be to fingerprint a given model of device. In [31] a theoretical model for user capacity is given. A

technique based on mutual information is given and demonstrated experimentally to predict how a system will behave with a large number of users using data from a smaller number of devices. The user capacity is specific to the set of features used, and estimating it requires data from a minimal number of devices.

In [45] hardware imperfections in the transmitter, the wireless channel, and the receiver are all examined as sources of variation. The effect of modulation type on fingerprints is modeled and simulated, although experiments are only performed for a singular modulation type. The simulations suggest that using a root raised cosine pulse shaping decreases the distinguishability of fingerprints, as does increasing the modulation order. Modeling the underlying causes of fingerprints is undertaken in [11, 51]. The contribution of the power amplifier to fingerprints is examined. Measurements and simulations are made with several power amplifiers. It is shown in [51] that an attacker's efforts to evade a PLI system by modifying their signal can be detected. The physical layer variation due to the imperfections in the DAC is also modeled in [11].

2.6.3 Fingerprinting for CRs and DSA

Features based on the second-order cyclo-stationary behavior of signals are proposed in [63]. These features should be robust to the time-varying noise and interference found in wireless channels, although it is noted they can be used in conjunction with other features to provide the best performance. In [44] the error signal between the received signal and the ideal is used as a feature. A deep convolutional neural network is trained to recognize or verify signals. Carrier frequency correction is used with the hope that the features learned by the neural network will be resistant to forgery. In a DSA network it is likely few users will have the high-end hardware typically used in the literature. Fingerprinting with low-end hardware is examined in [9]. Experimental results show that using lower end hardware for the fingerprinter causes fingerprints to become specific to the receiver used, and has a significant impact on performance. These works all examine possible applications to CRs DSA networks, but do not examine configuration dependency.

In [53] a transfer learning method is used with a PLI system to identify CRs. A transfer learning step is included to update the reference data for a CR each time its signal is verified. This compensates for small changes over time, rather than re-identifying a device after a large change as is examined here. In [64] a multi-view learning approach is used with location "fingerprinting" using outdated training data. The views are current and past measurements of RSS values for known points. A transformation between the past and present allows using the complete training data from the past time while requiring data from only a few training points at the current time. In this paper a similar multi-view approach is used, where each view corresponds with a specific CR configuration.

2.7 Conclusion

Physical layer identification provides a simple and backwards compatible method to help verify device identities. As increasingly varied applications are found for wireless transmitters PLI methods will need to adapt. Common attacks such as the Sybil and PUE attacks become substantially more difficult to detect when devices change configuration. Using state-of-the-art features based on the FFT it is impossible to recognize a device which has changed bandwidth or modulation type. Using the invariant distance metric proposed, devices can be identified with a high degree of accuracy. The method is simple, should not rely on any particular feature set, and has been tested with transmitter configurations including five modulation types, four bandwidths and two carrier frequencies. This allows current PLI methods to be applied to CRs in DSA networks using features with configuration dependency.

Based on these results, it would be useful to examine various hardware components as sources of the transmitter variation. It would also be useful to examine existing features for data dependency, as well as their performance with the proposed method. It has been shown in other works that the features in use here are very dependent on the CFO with the model of transmitter used. The results here showed that changes in carrier frequency do not cause configuration dependency, which suggests that CFO may provide an invariant feature.

Chapter 3

Crowdsourced measurements for device fingerprinting

Seth Andrews, Ryan M. Gerdes, and Ming Li

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

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.

Originally published in the proceedings of the 12th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2019).

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.

3.1 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 [47], including channel sensing [65, 66], and identifying the location of malicious users [67]. Incentive systems have been examined to encourage users to report measurements

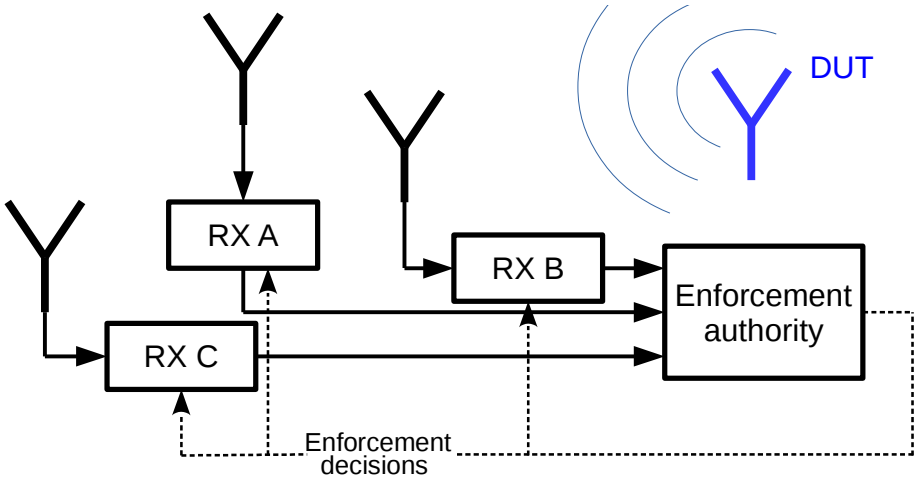


Figure 3.1: Diagram demonstrating a crowdsourced system. Three receivers capture observations of a signal and provide measurements (either the sampled signal or statistics extracted from it) to an enforcement authority to verify the transmitter's identity.

[68], and determine how to allocate a limited number of sensors to efficiently perform sensing [46].

We consider the step of identifying malicious users in such a system, shown in 3.1, 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.
- 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 3.3 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 3.5 we define combinations used for each level, and discuss how mismatch between receivers can impact low level combinations. In Sections 3.6 and 3.7 the experimental setup and results are presented for these methods as well as performance without crowdsourcing. We conclude in Section 3.10, including some possible extensions to this work.

3.2 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.

3.2.1 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 [69, 66].

We consider a DSA network, similar to that described in [69, 47], 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 [46], 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 [8], 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 [69, 67]. 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.

3.2.2 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 [2]. 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 [3]. 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 [70] 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.

3.3 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.

3.3.1 Fingerprinting works

A good overview of fingerprinting wireless devices at the physical layer is given in [2], as well as [3] 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 [2]. 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) [2, 43].

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 [70] 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 [10] 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 [70]), 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 [44]. 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 [71] 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.

3.3.2 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 [54]. 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 [66] . An overview of spectrum sensing is given in [72], 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 [8]. 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.

3.4 Preliminaries

Before examining crowdsourced fingerprinting, 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.

3.4.1 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:

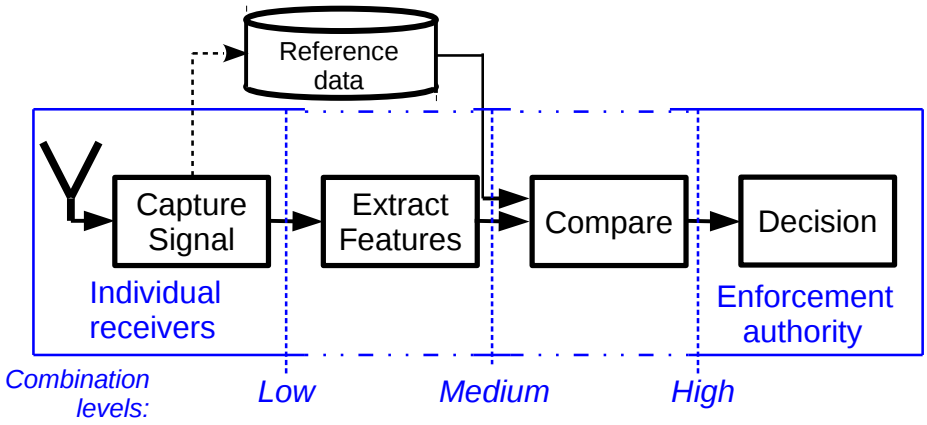


Figure 3.2: Steps in a typical fingerprinting system using a single receiver (black). Crowd-sourced approaches are marked in blue, with the responsibilities of individual receivers and the enforcement authority shown. Different combination levels are labeled in italics.

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 3.2. A variety of features have been used, as mentioned in Section 3.3. Frequency based fingerprints have been found to perform well, and provide a large set of features with good performance [73, 43]. We use them in this work, and describe them fully in Section 3.6.1. 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 [73]. Each possible 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.

3.4.2 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 3.2. 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.

3.5 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 3.7, 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 3.8.1, including further solutions to mismatch.

3.5.1 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 [71], 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[66]. 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.

$$V_H(y) = C_H(y^1, ..., y^n) = \prod_{i=1}^{n} d(R, f(y^i)) \tag{3.1}$$

This is the same form used in [44] 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.

3.5.2 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[2], this reduces the SNR as more observations are used. The resulting verifier is given by

$$C_M(y^1, ..., y^n) = \frac{1}{n} \sum_{i=1}^{n} f(y^i) \tag{3.2}$$

$$V_M(y) = d(R, C_M(y^1, ..., y^n)) \tag{3.3}$$

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.

3.5.3 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

$$V_L(y) = d(R, f(C_L(y^1, ..., y^n))) \tag{3.4}$$

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 3.5.3. 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.

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.

$$C_L(y^1, ..., y^n) = \frac{1}{n} \sum_{i=1}^{n} f(y^i) \tag{3.5}$$

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 3.7.

Nonuniform sampling algorithm

Nonuniform sampling algorithms provide an efficient and flexible approach to reconstruct the original signal[74]. Other approaches to handling nonuniformly sampled data are review in Appendix 3.9. 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

$$a = T^{-1}b \in \mathbb{C}^{2M+1} \tag{3.6}$$

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

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

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

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 [74]

$$y(t) = C_L(t; y^1, ..., y^n) = \sum_{j=1}^{r} a_k e^{2\pi ikt} \tag{3.7}$$

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

There are several things to note when solving (3.6) [74]. 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 [75] 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 [74], 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.

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 3.8.1. Equation 3.6 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.

3.6 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 3.7 are given. These include comparisons of low level methods, performance simulating several receivers without mismatch, and performance with actual mismatch present.

3.6.1 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 (3.6).

The high and medium combinations and the alternative low level methods in Section 3.5.3 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

$$f(y) = \left\{ F_k(y) : \frac{f_l N}{F_s} \leq k \leq \frac{f_u N}{F_s} \right\} \tag{3.8}$$

Typically the sampling rate must satisfy $F_s > 2f_c$ to prevent aliasing. However, subband sampling is used with (3.8) 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 [76]. 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 3.3, as well as the features extracted using (3.6).

3.6.2 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 [77]. 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 3.3, and the remainder discarded.

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

3.6.3 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

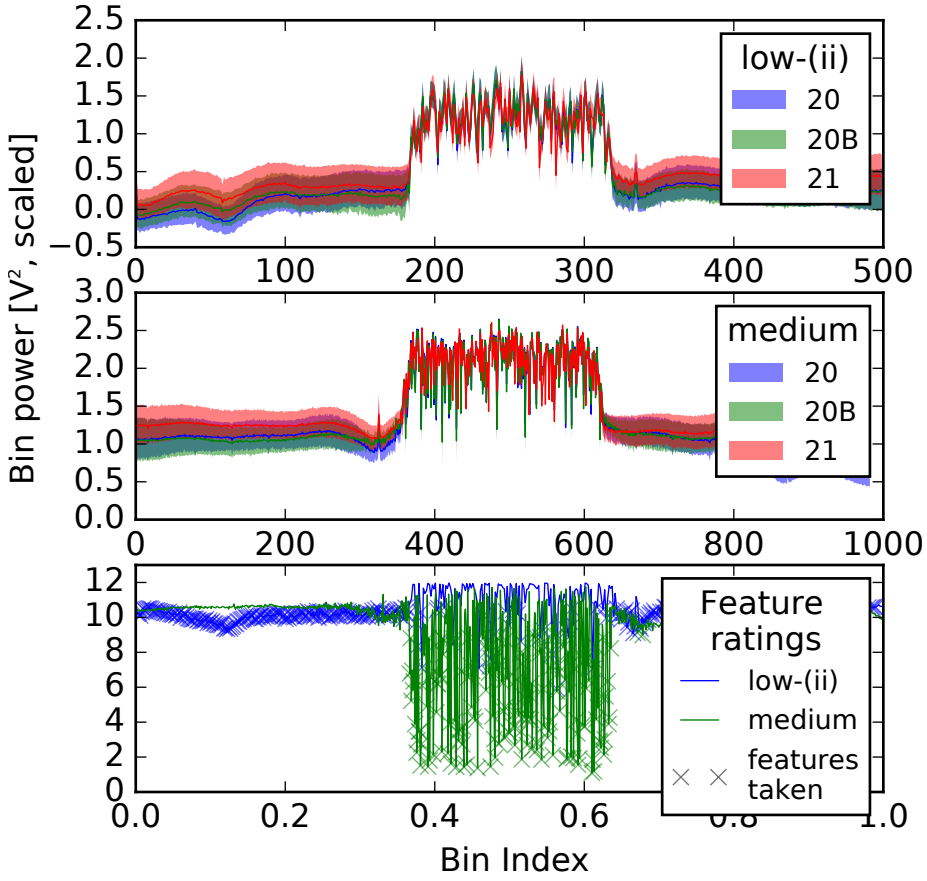


Figure 3.3: Example of features and feature selection. Upper: features extracted using Eq. (3.6), showing mean, quartiles shaded. Center: frequency features using Eq. (3.8). Lower: feature selection rating. There is some variation between the methods, and low level features favor features outside the main lobe.

than ideal performance for a single receiver, and demonstrate what improvements crowd-sourcing 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 crowd-sourced 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.

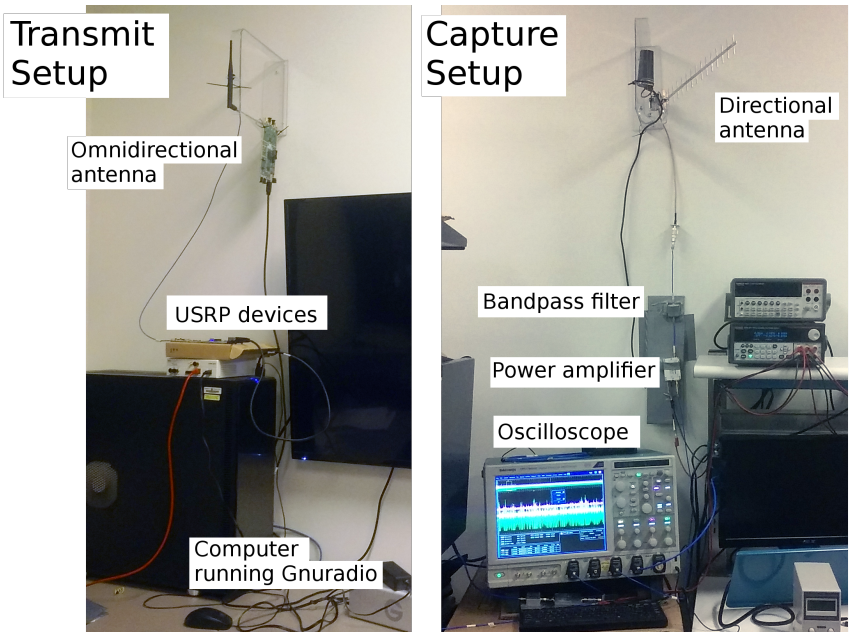


Figure 3.4: Data capture setup, showing transmitter and the oscilloscope with one receive antenna. Transmit and receive antennas are located on opposite sides of the lab.

Transmit setup

Ettus B210 radios [57] 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.

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 3.4. The layout of transmit and receive antennas is shown in 3.5. 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.

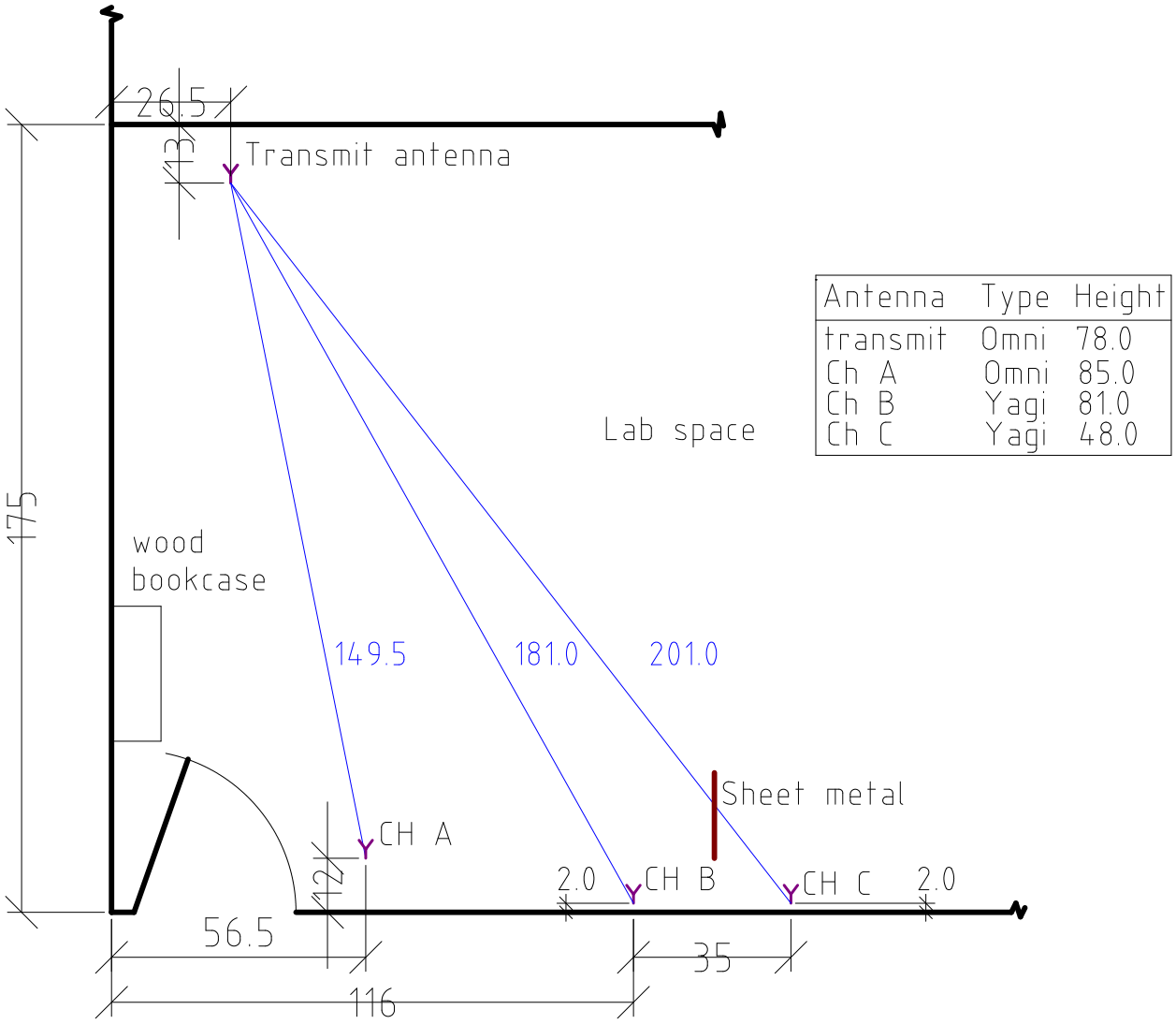


Figure 3.5: Diagram of antenna layout used. All dimensions are in inches, height is measured from floor. Tables, cabinets, other furniture and equipment below antennas are omitted.

The received signals pass through a bandpass filter and amplifier before being sampled by the oscilloscope [55] 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.

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.

3.6.4 Crowdsourced scenarios tested

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

Low level combinations

We consider five methods for low level combinations, based on those in Section 3.5.3 and 3.5.3.

i estimating frequencies using (3.6) when sample times are known exactly

ii using (3.6) with approximate sample times

iii averaging samples using (3.5)

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 3.6.1 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.

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.

Crowdsourced, with mismatch

The combinations are handled as in the previous section. We use observations from the three receivers described in Section 3.6.3. 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.

3.7 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 3.4.1. 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 [2]. For our applications, having a high TAR is most important, as this measures the impact the verification system has on legitimate users.

3.7.1 Individual receiver performance

Before describing the crowdsourced experiments, it's useful to present results without using crowdsourced data, shown in Figure 3.6. 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.

3.7.2 Crowdsourced performance

The results using crowdsourced combinations are now considered.

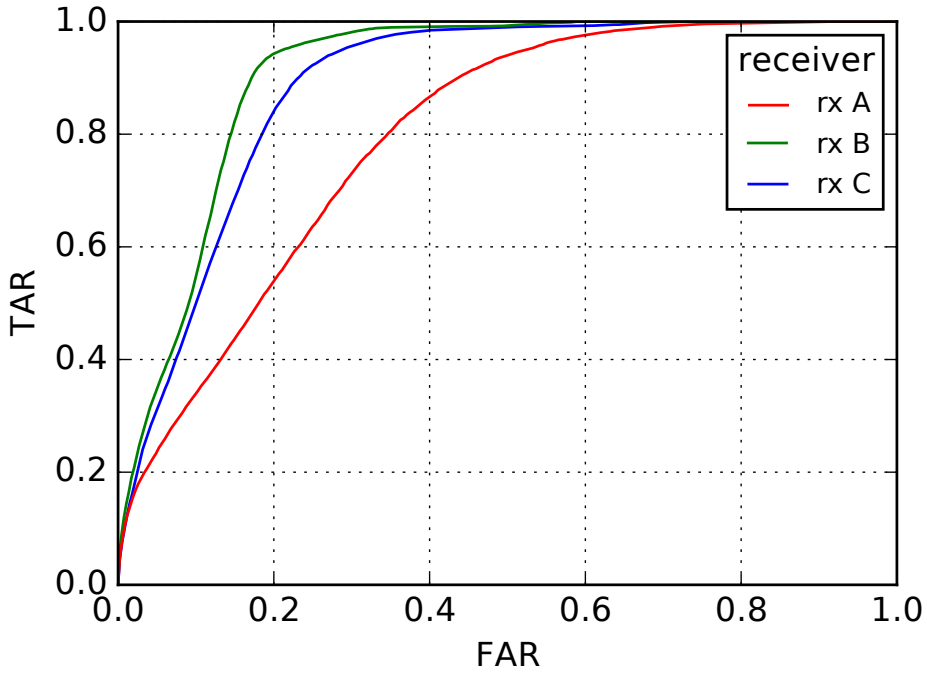


Figure 3.6: Performance of each individual receiver. The line of sight antennas perform best. The omnidirectional antenna results in a lower SNR and poorer performance.

Low level combinations

There is considerable variation between the low level combinations even when there is no mismatch between receivers, shown in Figure 3.7. 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 3.6 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 (eq:nusp) 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.

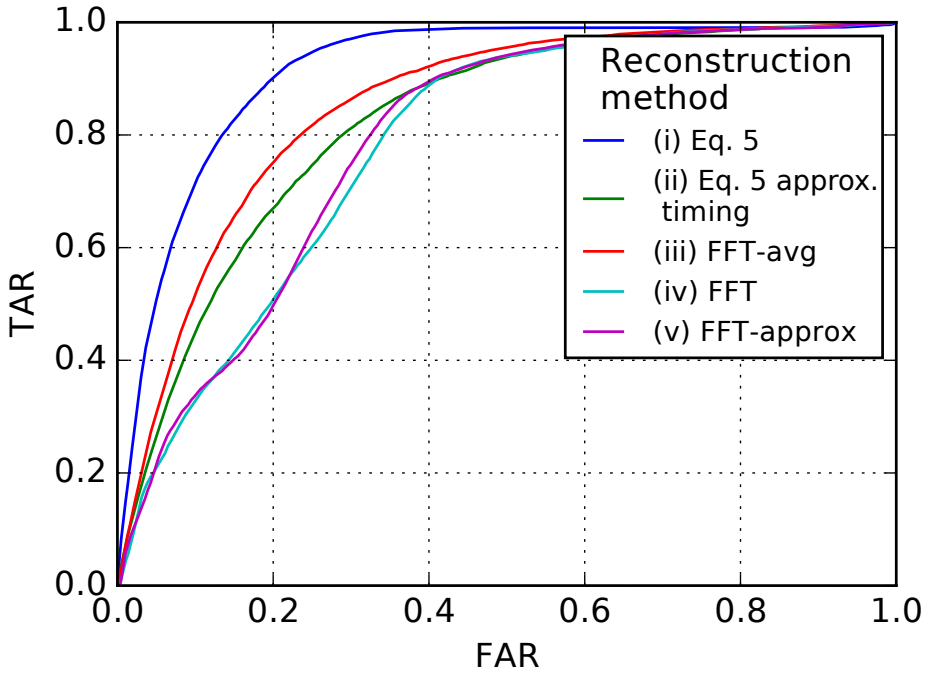


Figure 3.7: Performance low level methods, following the approaches outlined in Section 3.6.4. The approaches are based on whether sample times are known exactly or approximately, and whether the FFT directly or a nonuniform sampling algorithm is used.

Crowdsourced, no mismatch

With no mismatch, shown in Figure 3.8, 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.

Crowdsourced, all mismatch

With mismatch the results are substantially different, shown in Figure 3.9. 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.

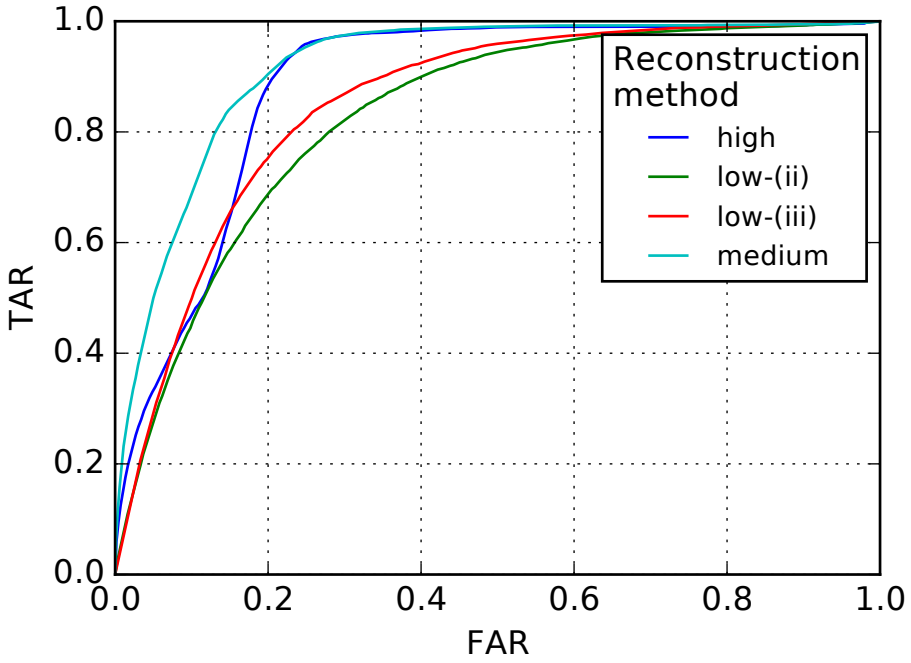


Figure 3.8: Performance of different combination methods with no mismatch. All methods perform well, with high and medium levels giving the best performance. Medium outperforms any individual receiver.

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.

3.7.3 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 3.1 for each method. Running time varies substantially between methods. Not surprisingly, the addi-

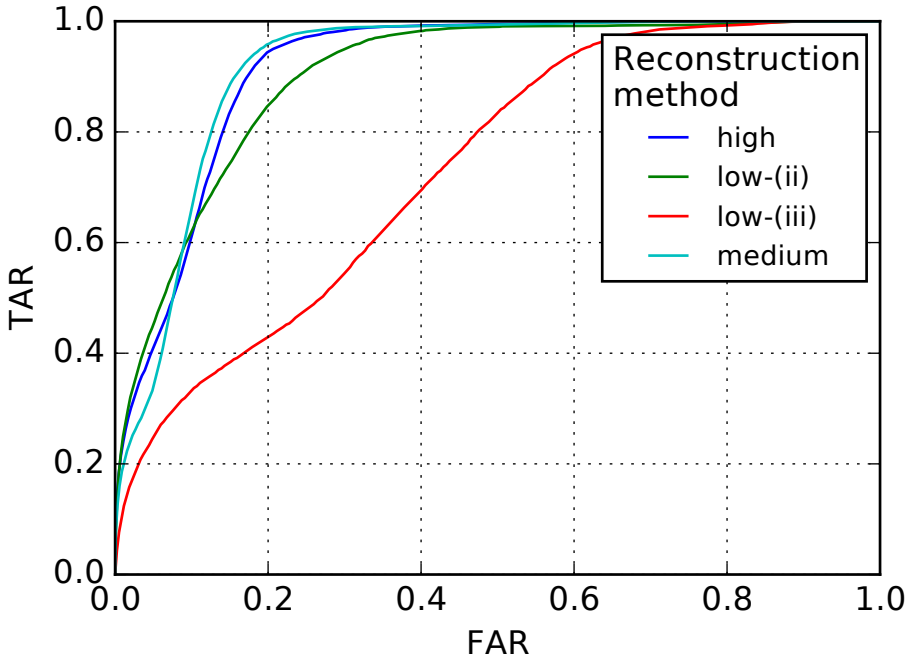


Figure 3.9: Performance of different combination methods with mismatch. Low-iii performs poorly, while low-ii improves in performance, for very low FARs it exceeds any individual receiver. Medium drops slightly in performance, but still outperforms any individual receiver.

tional 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.

3.8 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

	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	arbitrary
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 3.1: 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.

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.

3.8.1 Sources of measurement mismatch

The effects of offset, gain, timing, and bandwidth mismatch are described, based on the models in [78].

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 [78].

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 [78].

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.

3.8.2 Solutions to mismatch

There are a number of approaches to remove effects of mismatch. Solutions developed for interleaved ADCs typically have minimal overhead and real-time 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 homogeneous 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 [79]. 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 independently 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 [79]. 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.

3.9 Nonuniform sampling

We briefly review some approaches to handling nonuniformly sampled signals which are applicable to the low level combinations in Section 3.5.3. An early work is [80]. 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 [81]. 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 [82]. 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 [75]. 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 [74], which we describe in Section 3.5.3. 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.

3.10 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

Chapter 4

Sources of and solutions to drift in RF fingerprinting

Device fingerprinting has been proposed to help secure wireless networks in a variety of situations. Manufacturing variation introduces small differences between components, making every wireless transmitter unique. These differences manifest as slight variation in the signals emitted. Features based on these differences are extracted from a device's signal, forming a fingerprint which can be used to differentiate between devices. However, the features used for fingerprinting change with respect to environmental factors, such as temperature. This is called drift. Fingerprint features have typically been evaluated using data collected over a very small range of temperatures, and over a short period of time.

In this work we examine various transmitter components as possible sources of drift due to large and small temperature changes, find the impact drift may have on fingerprinter performance, and demonstrate some steps to improve performance. It was found that the oscillator is the primary source of drift due to temperature changes, with little or no contribution from the PLL, amplifier, and RF transciever (which contains the mixer and DAC on the transmitters examined). Performance is found using features based on the frequency content of the received signal. Changes in oscillator temperature above 8 °C were found to have a substantial impact on performance. This can be lessened by using features which are independent of oscillator behavior. However, we found that these features provide less ability to distinguish between devices. We also examine several regression models, and how they can be applied to predict drift due to temperature. Using these models, the effects of drift can substantially reduced if

Shortened manuscript based on this chapter is in submission.

the device's temperature is known, without sacrificing the ability to distinguish between devices.

4.1 Introduction

Device fingerprinting has received considerable interest, with proposals to use it to identify misuse of spectrum and even help authenticate devices in some communications systems. Since the hardware of a device does not change fingerprinting methods at the physical layer allow identifying devices, even when they change their identifiers. Drift occurs when device fingerprints change, typically due to aging of components or changes in environmental factors such as temperature. The performance of a fingerprinting system will obviously decrease when fingerprinting features change, whether due to drift or other factors. In addition to preventing identification, drift may cause the fingerprinter's actual performance to vary substantially from what would be predicted based on data without drift present. In wireless device fingerprinting direct control over the device under test (DUT) is typically not possible, so drift will occur in practice. For example, devices located outdoors experience regular temperature changes. Even in indoor environments, temperature can change throughout the day. Consequently, understanding drift is necessary to create robust and predictable fingerprinting systems.

A large variety of feature types and applications of fingerprinting have been discussed in the literature [2]. These include malicious WiFi access points, cloned RFID chips, and a variety of wireless transmitters. When features are examined for drift or changes it has been found that in limited experiments most common features do not change substantially over time, or with regards to small changes in temperature. A few works have looked at tracking drift in features [53]. While these work well, they require continuous observation of the device. In this work, we are primarily concerned with drift due to temperature changes. Drift due to changes in the channel can be compensated for with equalization, and drift due to changes in voltage, aging, and other factors are generally less than drift due to temperature. In the following, we use "drift" to refer to changes in device fingerprints caused by temperature variation, unless otherwise specified.

4.1.1 Contributions

A systematic study of the causes and effects of drift has not been undertaken. Recently, Pospíšil et al. [48] showed that large changes in temperature can have a substantial negative impact on fingerprinting systems. Here, similar issues related to temperature dependent drift are examined. Our contributions can be summarized as follows.

1. We examined transmitter components as possible sources of drift, and found that the

oscillator is primarily responsible for drift.

2. For the RF features used, we found that the carrier frequency offset (CFO) is a primary source of the ability to distinguish to between devices.

3. We demonstrate the use of several models, based on the behavior of transmitter components, to compensate for drift in features.

4.1.2 Organization

The remainder of the paper examines temperature related drift in several ways. First, a short review of the literature is presented, with a focus on fingerprinting works which examine features for drift, fingerprinting works proposing methods to track and compensate for drift, and works examining transmitter components as possible sources of drift. Next, the experimental setup is described including the features and classifier used, the transmitters examined, and how device temperature is controlled and measured. The experimental results are presented in three parts. First, several components are examined as possible sources of drift. The impact of drift on the true accept rate (TAR) and false accept rate (FAR) is quantified. Features not susceptible to drift are examined, as well as some regression models to compensate for drift. Lastly, conclusions are summarized, and some further work suggested.

4.2 Related work

Fingerprinting drift is a well known problem in the literature, although relatively few works address it directly. Some of the earlier works available describes specific emitter identification, and note the necessity of regularly updating reference data to account for drift [5]. Recent surveys have noted the importance of "stability" in fingerprints [4], and later works have attempted to circumvent the problem by looking for features with "permanence" or "robustness", which aren't subject to drift [2]. Here, works which describe the effects of drift, or test features for permanence are covered. Next, are those including steps taken to account for the effects of drift. Finally, a number of common transmitter components are analyzed for their susceptibility to temperature induced drift.

4.2.1 Impact of drift on features

Only a few features have been analyzed with regards to drift, typically to see that they are "permanent" with regards to voltage or temperature. Such features allow a fingerprinter to ignore the issue of drift. In [10] changes in voltage are examined, and found to have little

impact on features taken from fake cellphone base stations using software defined radios (SDRs). Device voltages in the range of 10.2 V to 13.3 V are examined, with reference data at 12 V. As the voltage increases or decreases there is a very slight drop in performance. In [43] smaller changes in voltage are found to have no effect. Temperature is also examined in [43], where changes of 5 °C are found to have no effect on performance. Larger changes were not investigated.

While it is simpler to limit applications of fingerprinting to those features and situations not subject to drift, this may not always be practical. Examination of the effects of temperature on drift has been done in [48] for some common features. These results are of particular interest as they use an SDR similar to the models used here. Specifically, the Ettus N200 with SBX and WBX daughtercard. We use the Ettus model B210. Features examined are CFO, IQ imbalance, origin offset, gain imbalance, mean power, and quadrature error [48]. Two transmitters are subjected to a temperature sweep from 0 °C to 50 °C in a temperature controlled chamber. In contrast, this work subjects each device to multiple temperature cycles, gaining insight into behavior dependent on past temperature. IQ imbalance and quadrature error are most impacted by changes in temperature, while gain imbalance is independent of temperature, and CFO exhibits a nonlinear and unpredictable behavior. Although not addressed in [48], the variation in the CFO may be due to temperature correction in the oscillator. Overall, the CFO changes by about 800 Hz, or on the order of 0.5ppm. The gain imbalance and origin offset appear to be unaffected. In terms of performance, changes of 10 °C are enough to cause misleading output from a fingerprinter, resulting in a high number of errors.

The effects of oscillator drift on features is mentioned in [83]. Phase noise introduced by the phase locked loop (PLL) is measured, using the autocorrelation of a captured record. This can uniquely identify transmitters oscillators. Two sets of features from eight oscillators are compared, with a three month interval elapsing between, and it is found that aging over this time has little effect on the features. Changes in temperature due to self heating are considered, and found to be small enough to ignore (on the order of 10s of Hertz). However, the oscillators are examined in isolation, rather than part of a transmitter's overall behavior.

4.2.2 Correcting for drift

If features are not "permanent" steps must be taken to compensate for drift. A few approaches have been taken in the literature, with varying suitability in different situations. These include updating reference data and modeling fingerprinter output. Updating reference data as fingerprints drift is the most common approach. This requires little knowledge about how or why features change, and a classifier that can be easily trained or updated. Continuous observations of the device under test are needed at some minimum frequency to ensure that it can continue to be identified [5]. If a device cannot be observed for a long period of time, such as when it is turned off, these methods may fail. A complete tracking

system is described in [5], including steps to update reference data periodically. "Insular isometric nets" are used as a classifier, which allow easily updating their weights, and a minimum update frequency is found for each cluster of features. In [53] transfer learning methods are used to update the reference data. Old fingerprints are down-weighted and eventually discarded. The benefit is shown to be primarily when a small amount of reference data is available. This is the case when limited observations can be taken, but may also be the case if features have a high rate of drift causing reference data to quickly become outdated. In [52] an upper and lower threshold is used in the comparison with reference data, and accepted records are used to dynamically update the thresholds. This accounts for small changes in device behavior, but requires a regular observation interval of the device.

Predicting or modeling drift is another approach, but has been investigated less. Features or fingerprinter output can be predicted based on time elapsed, device temperature, or other factors known to affect fingerprints. This allows re-identifying a device after a gap in observations. There is no requirement on a minimal time between observations, as with the previous methods. However, it requires having knowledge of the device's state (temperature, time since last observation, or other factors that affect drift), and accurately modeling all factors contributing to drift. The effect of temperature on the fingerprints of hardware keyloggers is modeled in [20]. Tracking drift using several time series models is undertaken in [84], including moving averages, auto-regressive moving averages, and several regression models. The fingerprinter uses matched filter output to identify Ethernet cards. Large and small changes in temperature are examined, ranging between approximately 28 °C to 62 °C and 28 °C to 31 °C. A number of modeling approaches are taken, but it seems that regression works best. The models are developed based on a single round of data collected, while here we develop models based on the behavior of transmitter components and test the models on additional rounds of data collection.

4.2.3 Sources of drift

Lastly, we look at possible sources of drift for some common components. The components that cause drift are the same components responsible for (at least some of) the variation used in fingerprinting, otherwise the fingerprints would not change. Consequently, several components identified in the literature as contributing to fingerprinting are examined, as well as common parts found in transmitters.

Resistors and capacitors are both impacted by environmental temperature. This is described by temperature coefficient of resistance for resistors. Datasheets for both resistors and capacitors will typically describe the change expected over a temperature range. Generally the change is quite small, although outside the rated temperature range it can grow unpredictably. Aging typically has a smaller effect than temperature on these components, see for example [85].

Amplifier behavior is impacted by temperature. Self heating in an amplifier can affect

fingerprints, but eventually amplifiers reach a steady state. This is generally not an issue in smaller amplifiers as their internal temperature reaches a stead state in a matter of tens of seconds [11]. For the transmitters used in this work, we found that the steady state is generally reached in one to three minutes. This is still a small enough time frame that it can be safely ignored, provided that the DUT does not shut off frequently. However, heating from an external source may still have an impact on amplifier behavior and fingerprints.

Besides CFO, oscillators may contribute to a number of fingerprinting features. Consequently, changes in oscillator behavior may have large effects on a fingerprinting system. Aging causes change in the behavior of crystal oscillators [5, 86]. They are also strongly affected by temperature, including temperature gradients across the crystal. Oscillator frequency can be affected by changes in air pressure, and can be permanently changed by impact and shock. The changes in behavior depend on the cut of the crystal and its turnover point. Oscillators also exhibit hysteresis for all but extremely slow changes in temperature [86].

4.3 Setup and experiments

The primary objectives of this work are to examine the sources of fingerprint drift, the impact drift may have on a fingerprinter's performance, and methods of compensating for drift. We consider a fingerprinting system which extracts features from the wireless signal to verify that the transmitter's identity is legitimate, and not an attacker impersonating another transmitter. For this reason, we also refer to the fingerprinter as a verifier. We are limited to examining drift caused by changes in temperature, as drift due to aging would require collecting data over a much larger time scale than is practical here. In practice, drift due to temperature is larger than that due to aging. For instance, the oscillator in the devices has a frequency stability of 2 ppm for changes in the rated temperature range. The stability is 1 ppm for aging in the first year, and 0.2 ppm for changes of 5 % in supply voltage [85, 87]. Consequently, it is expected that results found using drift due to temperature will encompass results due to aging and other factors. Temperature is also straightforward to analyze in a limited lab environment. Changes in the ambient temperature as well as the temperature of specific components on the board are examined. Voltage has been examined in other works, and found to be minimal for the amounts of variation expected in typical operation [10, 43]. The features used are based on the fast Fourier transform (FFT) of each signal (also referred to as power spectral density (PSD) based features in some works), in contrast to the modulation based features used in [48]. These have been used extensively in Chapters 2 and 3, but have not yet been analyzed with regards to drift.

The experiments are rather simple, and focus on a limited yet complete analysis rather than superficially testing a large number of devices, components, and other variables. The setup is used in the following sections to determine how components contribute to drift, what the effects of drift are, and how to compensate for drift. The datasets gathered will be described more fully where they are used in Sections 4.4, 4.5, and 4.6. Here, the experimental setup

used to gather data is described, as well as the features and fingerprinting techniques used. This is followed by a more in depth description of the devices used, and how temperature is controlled and measured.

4.3.1 Fingerprinting setup

The fingerprinting setup is largely the same as that used in [49, 1]. Consequently, only a short description of the key points is given here.

Ettus model B210 SDRs are used as the DUT. Each DUT transmits using the same antenna in a fixed position to remove channel effects and ensure the features originate from hardware variation. GNU Radio is used to create the IQ waveform at baseband, using the same bit sequence for all devices. This solves the problem of data dependency which occurs in the features used. In a real world setup the synchronization bits or another portion of the signal that is constant across all messages could be used. Some minor changes were made to the transmission program used previously to save temperature data. The DUT modulates the baseband signal with the carrier frequency and transmits. On the other side of the room, at a distance of 4.6 m, a receiving antenna feeds into an amplifier, then the oscilloscope [55] captures the signal at a rate of 25 GHz, and saves it for analysis. Features are extracted from each record captured this way.

4.3.2 Devices used

The devices tested are four model B210 SDRs, from Ettus Research [57]. These allow for full duplex transmission on two separate frontends, using an RF transceiver chip from analog devices [88]. Here, the complete transmit frontend is described with the objective of understanding the contributions of each component to the output. We are not concerned with the receiver portion.

There are a number of components of interest in the receivers, depicted in Figure 4.1. The key part of each board is the chip implementing the RF transceivers. An FPGA handles some digital processing of the IQ data for each frontend, which is sent to the chip. Each board has a single oscillator, driving a PLL which is then fed into the RF transceiver. The transceiver chip synthesizes the final carrier frequency, and handles mixing this with each baseband signal then outputting to a digital to analog converter (DAC). This is done independently for each frontend. Each frontend also contains independent filters and a small amplifier after the DAC. The output of the RF chip for each frontend is then fed through a slightly more powerful amplifier and output. The board is rated for 25 °C, and other boards in the same family recommend a temperature range of 0 °C to 40 °C and −40 °C to 75 °C with an appropriate enclosure.

Although each board contains two independent frontends, and these frontends only share a

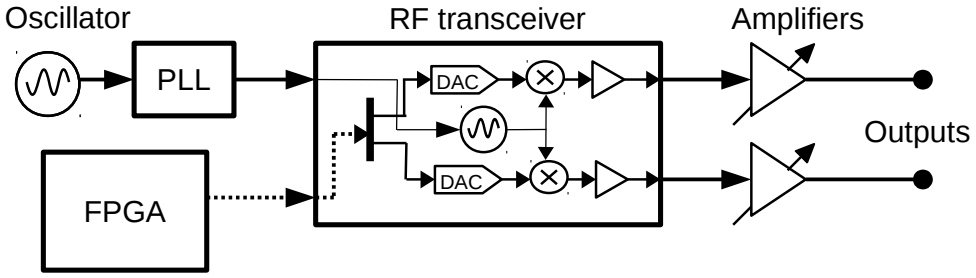


Figure 4.1: Diagram of the boards used, showing how different components contribute to the output of each frontend. Does not reflect actual placement on board. The RF transceiver handles signal generation for both frontends, which can be amplified further by external amplifiers. A single oscillator generates the frequency for the entire board.

few components, their behavior is very similar. This is examined in Section 4.3.2. Due to this similarity between transmitters located on the same board only a single frontend is used from each board in the following experiments. This reflects a more realistic distribution of hardware variation among the devices tested. By doing this overall performance improves in all cases, however, the effect on the overall conclusions is not large. The improvement in performance is substantial when no correction is taken for drift, and when regression based models are applied. It is a smaller improvement for the features based on the signal envelope. However, the relative performance of the different methods does not change noticeably when using only a single frontend per board.

Features

The features used are based on the amplitude of the FFT coefficients of the signal, following the descriptions given in [1]. They are state of the art, following the features used in [43, 89], and similar frequency based features have been used in the literature, including [90, 42], [9]. Before extracting features the signal is decimated to a rate of 100 MHz. This downconverts the frequency content, and allows for a lower sampling rate and faster processing. Bins centered around the aliased carrier frequency are selected as features, containing both modulated data, sidelobes, and the noise floor. These features reflect the frequency response of the transmitter's frontend, and other components. However, it was found that they are also very sensitive to CFO. Consequently, results using features based on the frequency content of the envelope rather than the modulated data will be shown later. Fisher feature selection is used to find the 250 most discriminating features, which are then used for fingerprinting.

These features are shown in Figure 4.2. The features are extracted from records taken from four transmitters on two separate boards. Section 4.3.2 describes these in more detail, including which components are unique to each transmitter and which are shared on a board. It can be seen that transmitters on different boards have quite different fingerprints, while

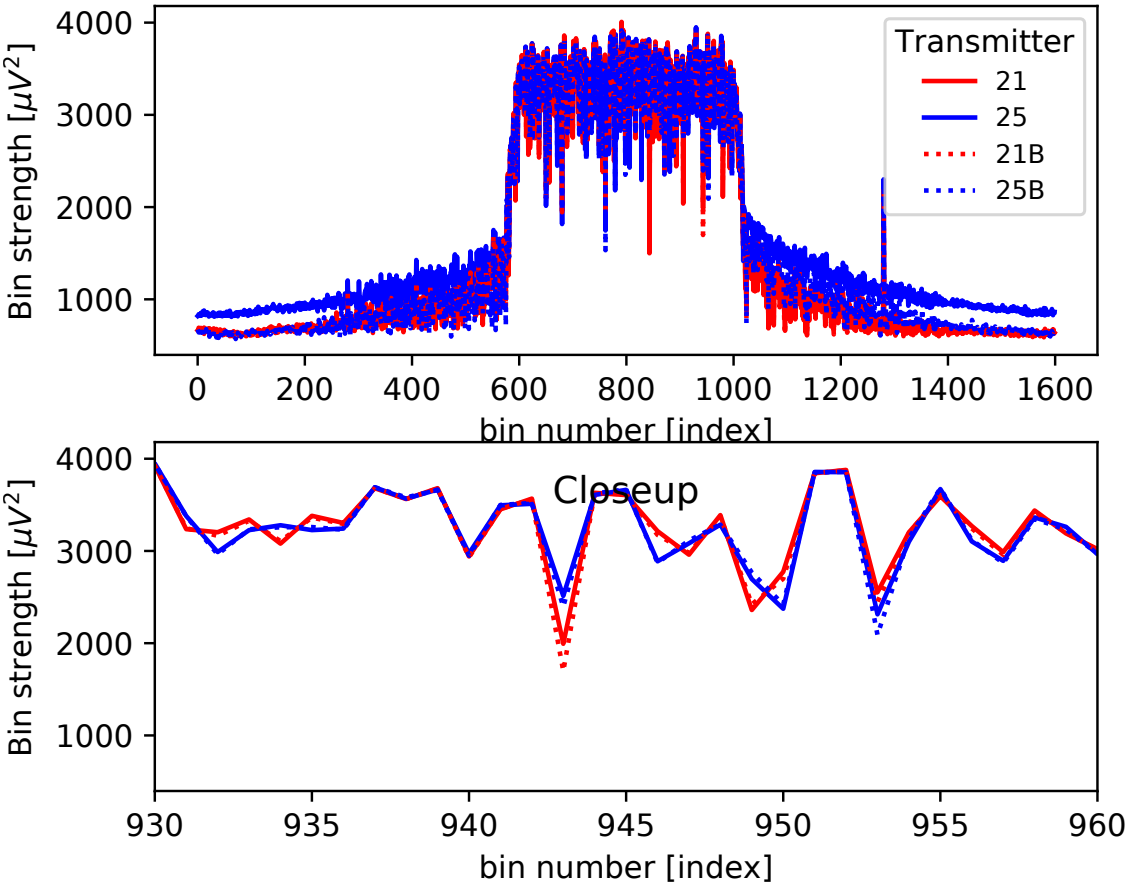


Figure 4.2: Features used, averaged across all records for four transmitters on two boards (21 and 21B are on the same board, as are 25 and 25B). The features are extracted from records collected at a constant temperature. There is some difference between transmitters on different boards, but very little difference between features from different transmitters on the same board.

transmitters on the same board appear very similar. This is in contrast with the results in our previous work [49] which found that frontends on the same board behaved differently, and consequently treated each as a distinct device. The difference here is most likely due to using a consistent temperature when collecting reference data. Self heating can alter device behavior, and differs depending on the frontend used. The different frontends will heat different portions of the RF chip, leading to different behavior. When components besides the RF chip are cooled to a uniform temperature transmitters on the same board behave very similarly.

Verifier

The fingerprinter (or verifier) compares the DUT's signal to reference data from the identity of interest. Mahalanobis distance is used, as described in Chapter 2. Other metrics can be used, or classifiers such as k nearest neighbors, neural networks, or support vector machines. This comparison is done with a record from the DUT and reference data from the identity to be verified. The DUT's identity is accepted when it falls within a given threshold of similarity. This threshold is chosen based on the false accept rate and true accept rate. The TAR describes the number of true positives, or the percentage that the verifier correctly accepts the DUT's identity. The FAR describes false positives, or the rate at which the verifier accepts a device's identity when it is incorrect. The threshold expresses an operator defined trade-off between these two rates, and can be described by the receiver operating curves. This curve is found by plotting the TAR versus the FAR for all possible thresholds. An ideal curve has high TAR for all FARs, and a low FAR for all TARs.

Mahalanobis distance is used to find the similarity between the reference data and the DUT's signal. Mahalanobis distance is proportional to the likelihood that DUT's record came from a Gaussian distribution with parameters derived from the reference data. When the DUT's identity is correct, this distance will be very small. Other devices will generally result in a large distance. Performance using these features is given later. Next, the transmitters used are described, followed by methods to control temperature.

4.3.3 Temperature measurement

First, the equipment for adjusting temperature is described. This enables two approaches to adjusting board temperature. Then, the methods of measuring the temperature of each transmitter and individual components on each transmitter is described.

The entire transmitter could be heated or cooled using an air conditioning (AC) unit. The AC unit feeds into a partially enclosed box, shown in Figure 4.3. This limits the area heated or cooled, and minimizes changing air currents. This doesn't have exact temperature control and the enclosure limits how precisely it could be set, but sensors on each transmitter will show the actual temperature. The temperature was set to hot (31.1 °C), and cold (22.8 °C), and changed directly between these. It could be set to intermediate temperatures, but this was not done in practice. Although this was the temperature set in the AC unit, the temperature in the enclosure could in practice exceed these values as it is much smaller than what the AC unit is rated for. When the board is heated this way drift cannot be attributed to a single component – the device's output is the result of all components being heated.

Second, individual components on each board can be heated. The heating element is a soldering iron at the lowest temperature setting, with a wide chisel-like tip. The wide tip maximizes heat transfer, and is aided by ample thermal paste being applied between the heating element and component being heated. A constant temperature is maintained

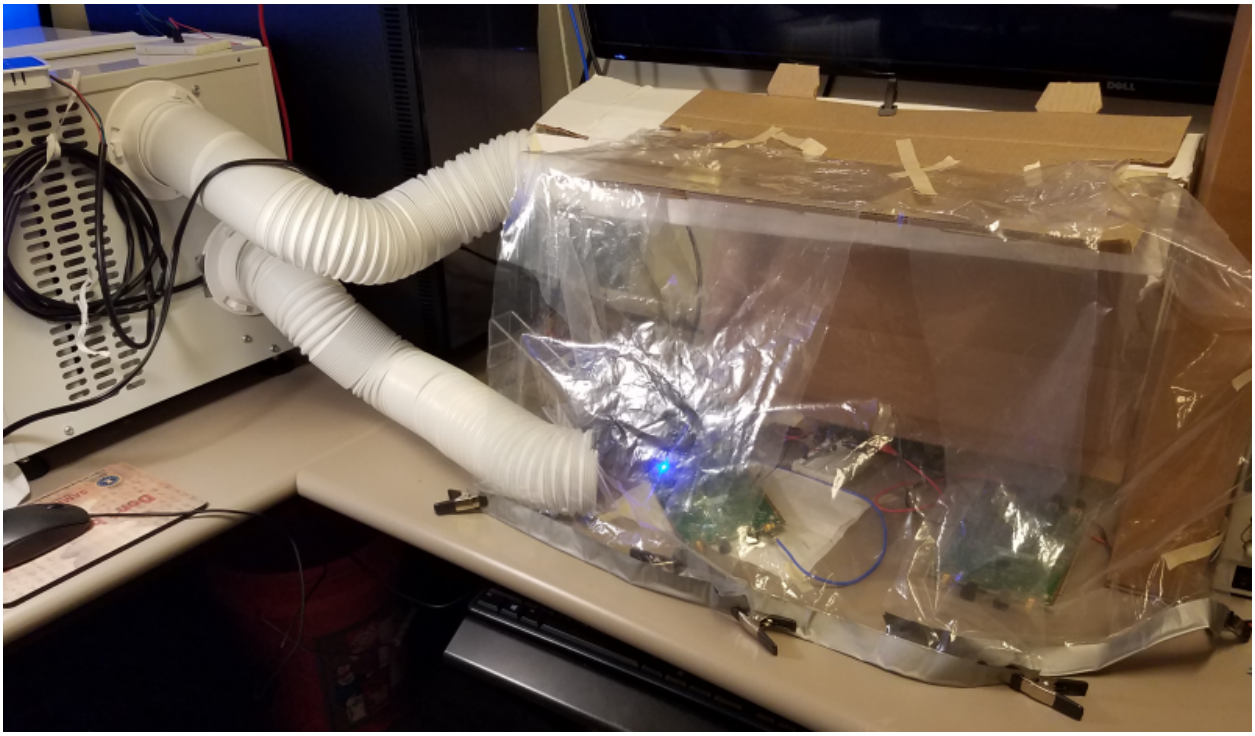


Figure 4.3: AC unit and the enclosure used to help regulate temperature. Transmitter is shown without heating element or any external temperature sensors attached.

over the entire board using the AC unit while individual components are heated. Heat is applied or not, intermediate temperatures are not possible. In initial experiments the heating element was moved between components. However, opening the enclosure and moving the iron was found to introduce as much, or more, variation than heating the components tested. Consequently, the heating element is placed on a single component beforehand and turned on or off using an external switch so no alterations are made to the setup.

Two types of sensors are used to record measurements. First, each transmitter includes a temperature sensor in the RF transceiver. Specifications are unavailable, but it tends to provide a very noisy measurement. Consequently, 100 measurements are taken over 1 second and averaged. the resulting temperature is for only a single component, but when entire board is heated this temperature should closely reflect all components. Second, in some experiments external temperature sensors are attached to individual components. The external sensors take the average of 2500 measurements over 0.25 s to reduce noise. The sensors used are accurate to within around 0.5 °C, over a much wider range of temperatures than those used in the experiment [91]. The measurement reflects an intermediate temperature between each component's temperature and the ambient temperature. The components measured are the amplifier, RF transceiver (in addition to the internal sensor), PLL, and oscillator. Most components are too small to accommodate both the temperature sensor and heating element – only the RF transceiver has enough surface area to accommodate both. The other components have a small aluminum slug clamped over the element with thermal paste in the connection. This provides enough space to attach the temperature sensor and heating element. This should have a minimal effect on measurements, as it easily conducts heat. An additional benefit is that this prevents the thermal paste used with the heating element from leaving residue on the board.

Two types of experiments were conducted: 1) where the board was cooled to a constant temperature with components on it selectively heated, and 2) where the entire board was heated or cooled. When the temperature of individual components is altered external temperature sensors are used. When the entire board is heated only the internal measurement is used . Some additional steps had to be taken to ensure consistent measurements, unless otherwise specified. The oscillators are very sensitive, even placement of thermal paste to attach temperature sensors was found to affect their measurement. Minimal thermal paste was used, and once the temperature sensors and aluminum slugs for measurement had been placed several runs of data were taken with the same setup. Placement of the board in the enclosure, especially in relation to the air intake, could affect readings. Efforts were taken to place each board in approximately the same position, and the setup was disturbed as little as possible during each run.

4.4 Sources of drift

The main objective of this section is to determine which components of the transmitter are responsible for fingerprint drift, as caused by temperature. Specifically, the contributions of the RF transceiver, oscillator, amplifier, and PLL are examined. Resistors, capacitors, and other small components were not considered. The experiments commenced with the larger components listed, and found these were primarily responsible for changes. Based on this, and the expected impacts of temperature on these components as covered in Section 4.2, investigation of other components was not done.

On a high level, the approach to experiments taken in this section is to selectively heat each component and see what drift occurs. In the following drift is measured in terms of change in verifier output. As the fingerprints drift, the verifier's output will increase in magnitude, due to the increasing difference between the DUT's signal and the reference data collected from it at an earlier time. As an initial analysis, the heating element is applied to each component for a few minutes so that it reaches a high temperature, then allowed to cool before heating the next component. Results are shown in Figure 4.4, for the four components considered. Typically results of a verifier's output are shown using a single set of reference data with multiple sets of test data. Here a single set of test data is shown with distances to different sets of reference data. This works better with changing temperature because the test data changes inconsistently between devices while the reference data is nearly constant. Consequently, by showing distances to multiple sets of reference data substantial changes in fingerprinter output are visible and clearly attributable to a change in temperature.

For most of the data in Figure 4.4 there is a clear separation between transmitter 24 and the other two transmitters. When the amplifier, RF transciever, and PLL are heated there are only small changes in distance, and these appeared to be due to disturbances from moving the heating element, rather than actual heating of individual components. This is good - it means a low FAR and TAR is possible, corresponding to a threshold distance between transmitters 24 and 23. However, when the oscillator is heated (around minute 3) the transmitters become indistinguishable: the distance to the DUT increases, while the distance to the other sets of reference data (transmitters 20 and 23) decreases. For most of this time the distances to reference data from the DUT overlap with all other reference data, preventing identification. This shows that the oscillator is the biggest contributor to drift, and will prevent fingerprinting.

While this shows approximate causes of drift, a more precise method is needed to accurately determine each component's contribution. As mentioned, some changes in distance may be due to moving the heating element between components, or to air flow through the enclosure after it is opened and closed to move the heating element. The oscillator is also very sensitive to environmental behavior – even residual thermal paste can affect its behavior. Additionally, measurements of the temperature sensor can change depending where it is placed on larger components, such as the RF transciever. Placement of thermal paste and the heating element

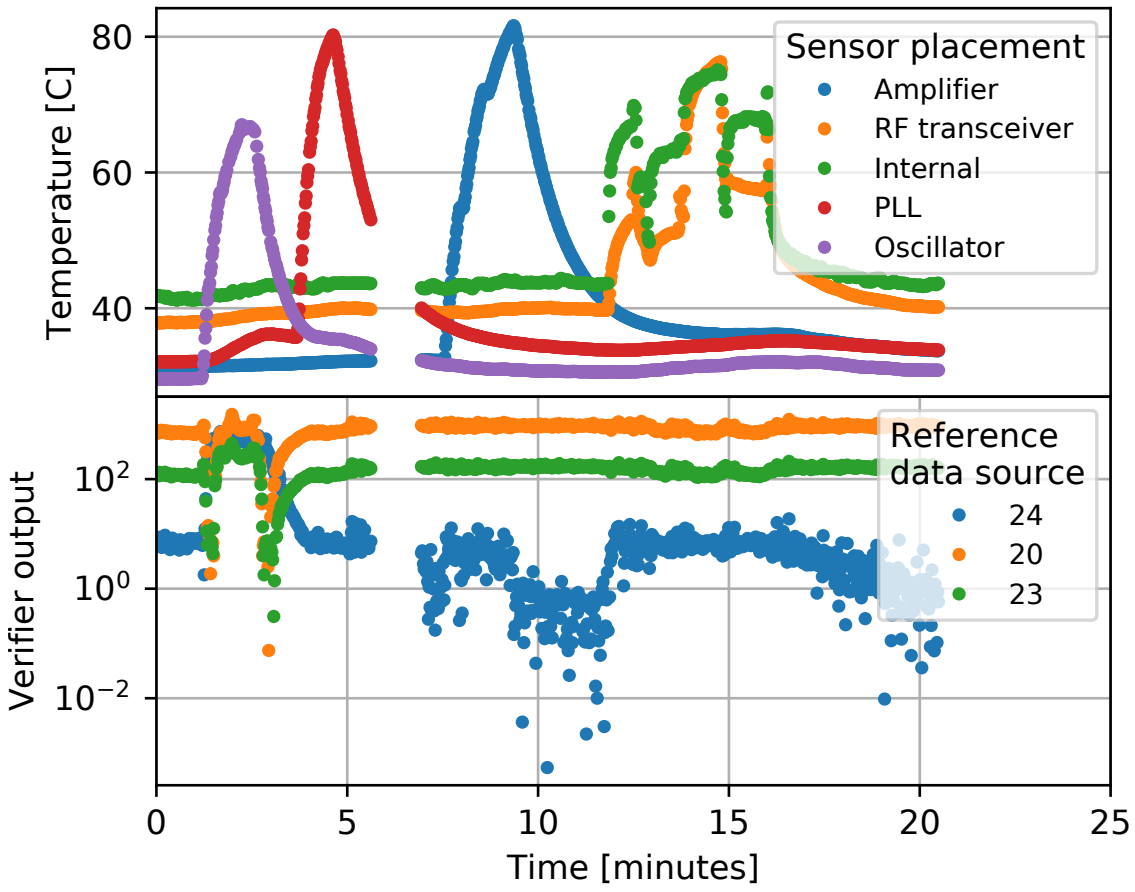


Figure 4.4: Results of selectively heating components on transmitter 24. The temperature of each component tested is shown (upper) with the corresponding fingerprinter output normalized using typical distances from transmitter 24(lower). The gap at 6 minutes is due to the capture setup's amplifier becoming disconnected. Components are heated one at a time, so some small differences in distance may be due to moving the heating element. The fingerprinter output shows that the oscillator has the largest effect, with a smaller effect from the RF transceiver and little or none from other components.

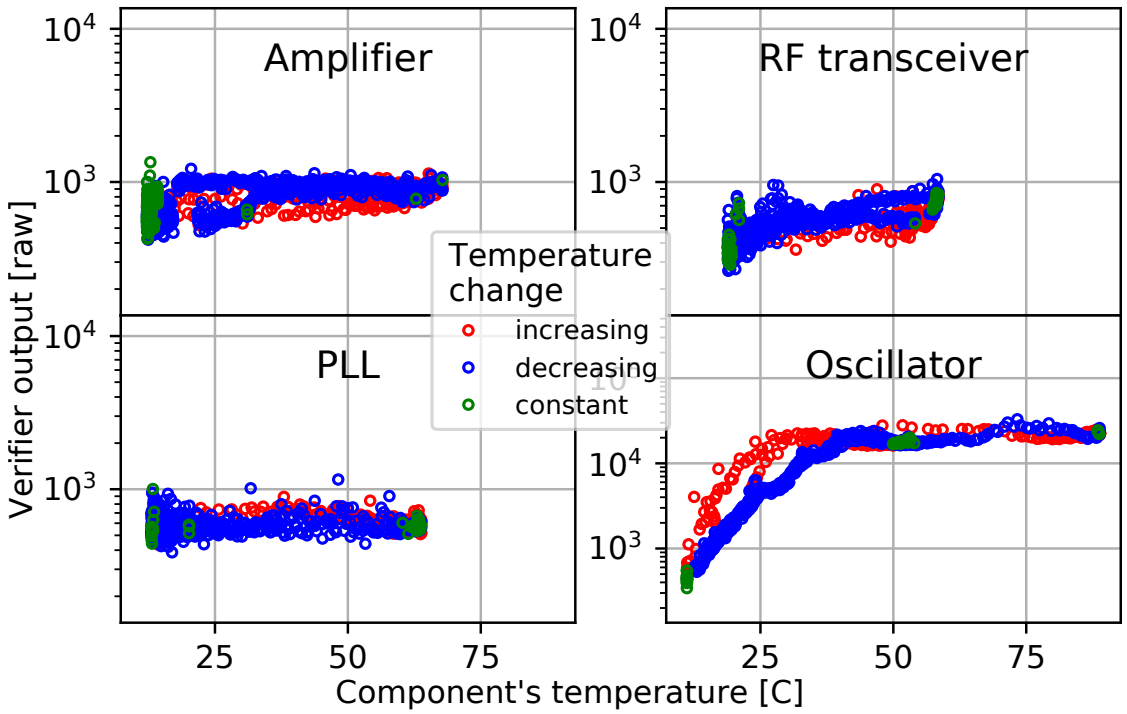


Figure 4.5: Correlation between each component's temperature and fingerprinter output, when each component is heated and cooled repeatedly. The points are colored according to whether the temperature is increasing or decreasing. The oscillator has by far the largest impact, and exhibits significant hysteresis. The other components exhibit this to a much lesser extent.

could also alter behavior, albeit in minor ways. To counteract these effects, in the following experiments sensors are adjusted as little as possible once they are placed, and a minimal amount of thermal paste is used on each component. The heating component is placed in single location for an entire run (collecting data from one transmitter with one component heated), and can be turned on or off from outside the enclosure. This prevents any changes in air flow throughout the run. Only a single component can be tested, but all variation should be due to temperature. Heating one component can affect nearby components, although only the PLL and oscillator are located in very close proximity. This was minimized by cooling the entire enclosure to a constant temperature at the same time that individual components are heated. This allowed for determining individual contributions with a high degree of accuracy.

Results of heating each component while taking these precautions are shown in Figure 4.5. These, and all other distances, are normalized using the mean and standard deviation of distances from all transmitters when operating at a constant temperature. The reference data used to normalize the distances was collected without temperature sensors attached. The normalized distances are still very large, which demonstrates the effect temperature

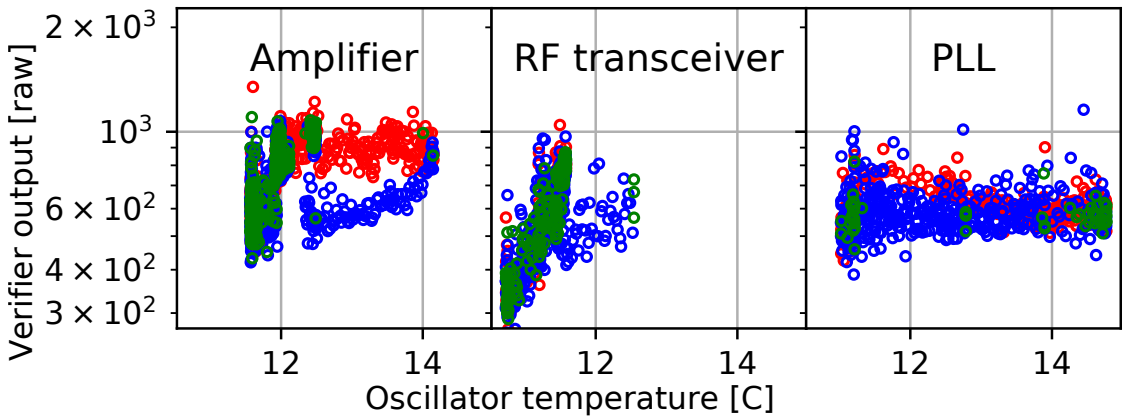


Figure 4.6: Correlation between oscillator temperature and fingerprinter output, for the same data used in Figure 4.5. The oscillator temperature explains most of the variation in the amplifier and RF transceiver (aside from a few outliers). The PLL still has very little variation.

sensor placement has. Amplifier temperature and fingerprinter output has some correlation. They appear to be most correlated for decreasing temperatures above 30 °C. Heating the RF transceiver leads to larger verifier output, but the correlation is weak. Some hysteresis is visible, with increasing temperatures having slightly lower verifier output. The PLL shows minimal correlation. The most substantial changes in output occur when the oscillator is heated. The verifier output changes by several orders of magnitude, but above 50 °C remains relatively constant. Hysteresis is clearly visible as the output changes, with increasing temperature resulting in a higher verifier output. This matches the expected behavior of a heated oscillator, as their frequency is known to drift and exhibit hysteresis when heated [86]. We found no evidence that the other components examined exhibit hysteresis.

The oscillator provided the greatest change in verifier output. Consequently, it makes sense to re-examine the contributions of each other component in terms of the oscillator temperature. The subplots of Figure 4.5 are reproduced in Figure 4.6, using oscillator temperature rather than each individual component. These are the same distances, but showing the correlation with oscillator temperature rather than the component being heated. For the amplifier and RF transciever there is a strong relationship between oscillator temperature and fingerprinter output. The oscillator temperature exhibits a stronger relation than using the component's own temperature did, excepting the PLL which is unaffected in both cases. Care was taken to minimize heat transfer between components However, it was observed that even small changes in the temperature of the oscillator are sufficient to cause significant changes in verifier output. This suggests that, not only is the oscillator the strongest contributor to drift, it is responsible for the majority of the drift when other components are heated. Even a small change in oscillator temperature can have an effect greater than a large change in the temperature of other components.

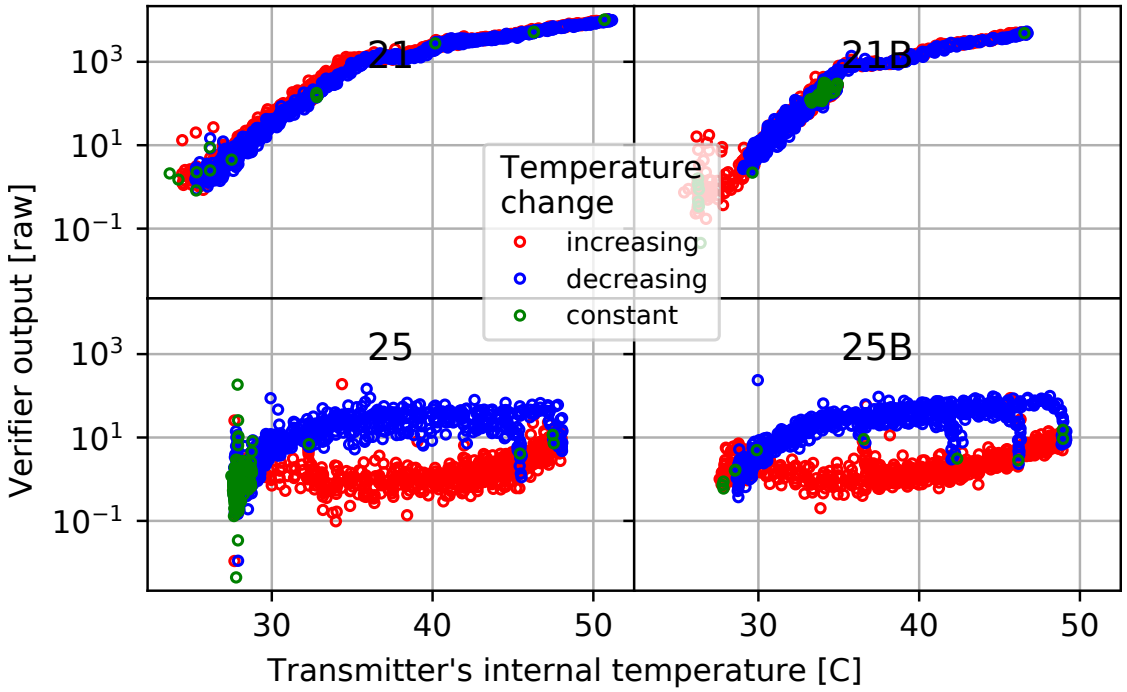


Figure 4.7: Correlation between each transmitter's internal temperature and fingerprinter output. Each transmitter is shown in a separate subplot. Transmitters on the same board (21 and 21B, 25 and 25B) exhibit behavior more similar to each other than transmitters on different boards: board 25 exhibits significant hysteresis but relatively little change in distance due to temperature, while board 21 has little hysteresis and substantial changes in fingerprinter output.

Lastly, the relationship between temperature and fingerprinter output is compared to see how it varies between different transmitters and boards. The relationship between the internal temperature of the RF transciever and fingerprinter output is shown for four transmitters on two boards in Figure 4.7. Each transmitter is put through several temperature cycles between 10 and 20 minutes in length. The entire board is heated and cooled for these experiments, so the internal temperature should be highly correlated with the oscillator temperature. Both transmitter frontends on each board are tested, for a total of four transmitters. Since transmitters on the same board have some common elements there is behavior that is similar across transmitters on the same board, but different between transmitters on different boards. The amount of hysteresis, and amount of change in distance per change in temperature is common to devices on each board, although there are slight differences in behavior.

4.5 Impact on performance

Clearly, fingerprint drift will make it more difficult to identify devices. It causes the DUT to appear dissimilar from its own reference data, decreasing the true accept rate (TAR). It may also causes other devices to appear more similar, increasing the false accept rate (FAR). Here, we try to quantify the impact this may have. The experiments use data from four transmitters. This is a limited number compared to many fingerprinting works, but we are interested in the impact on performance changes will have, rather than the overall performance of a practical system. Since boards are affected similarly by changes in temperature, it is expected that the effect on performance will accurately reflect the impact on a system with a larger number of devices.

The reference data is collected while the transmitter is cooled at a constant temperature (around 27 °C measured by the internal sensor). While collecting the test data the DUT is put through several heating and cooling cycles, each lasting for around 10 to 20 minutes. Each transmitter experienced changes in temperature of around 20 °C overall. The minimum and maximum temperatures for each run are shown in Table 4.1. Around 1200 records are collected for reference, and 1200 for test from each device, although this varied slightly between runs. The internal temperature sensor is recorded and used to separate data into several sets based on the change from the reference data's temperature. When the entire board is heated the internal temperature sensor will be highly correlated with oscillator temperature, which is the primary source of drift. It is also a measurement available to SDR devices without modifications to hardware.

Two sets of data were taken for each device, several days apart, to help validate the models considered in the following section. Only a single run from each transmitter is used in this section. Actual values for one run are shown in Figure 4.8, with accompanying fingerprinter output for reference data from transmitter 21. The distances are normalized, in this and the following figures showing distances, so that each unit distance represents one standard deviation from typical reference data. The distances for the DUT are heavily impacted by temperature, and change by over 10 standard deviations for the DUT, while the impact on other devices varies by greater or lesser amounts. For some temperature values the devices are indistinguishable.

Results are also presented in terms of receiver operating curves, in Figure 4.9. One transmitter is taken as the legitimate device, with all other devices treated as an attacker. This reflects a realistic scenario where the ambient temperature around the DUT changes, as well as the ambient temperature around attackers and other devices. This is repeated for each transmitter as a legitimate device, and the resulting distances used to find the receiver operating curves. The receiver operating curves express a trade-off between the TAR and FAR: as the TAR increases the number of false accepts must also increase. An ideal curve would allow a TAR of 1, with no errors made (a FAR of 0). This corresponds to a curve in the upper left. Results are grouped, with each record assigned a group based on the absolute

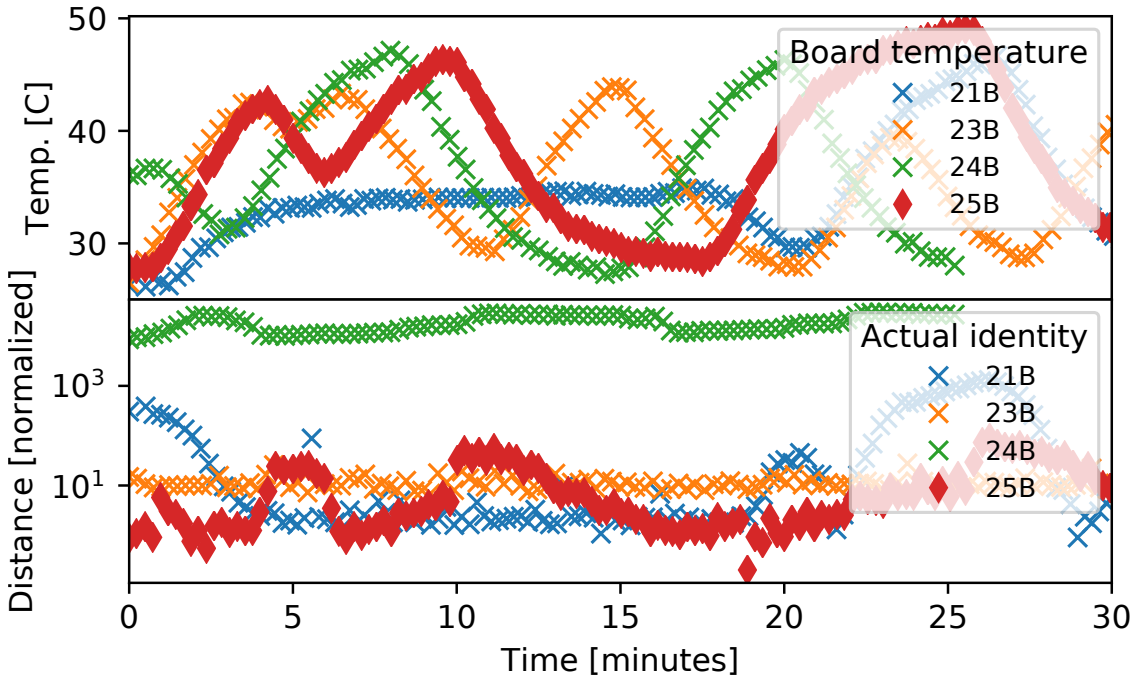


Figure 4.8: Internal temperature of each device as it is heated (upper), with corresponding fingerprinter output (lower). Reference data from transmitter 25B is used, and distances are normalized using the mean and variance of distances from all transmitters. It can be seen that when large changes in temperature occur at the DUT identification is not possible. The other devices are affected by changes in temperature in varying ways.

difference of its temperature from the reference data's average temperature. Since the reference data was collected while cooling, generally the test records have a greater temperature. In a very small minority of cases their temperature is less. When the change in temperature is less than 2 °C performance is not perfect, but the curve is near the upper left for most FARs. Larger changes decrease performance, temperature changes above 8 °C leave the verifier completely confused about the identity of some devices. In [48] changes of 10 °C or more were observed to have an impact on system performance, with larger changes preventing identification entirely. This is a larger temperature than what was found here, however the ambient temperature around the board was used while here we use the measurement of the internal temperature of the transceiver chip. Additionally, [48] measures performance with five devices at room temperature and the sixth experiencing a change in temperature. We vary the temperature of all test data, which may have a greater impact on performance.

Table 4.1: Temperatures covered by data run for each transmitter, in degrees Celsius. Different temperature ranges were covered to help show that the model can regression models found can generalize outside temperatures seen. Each transmitter had data collected at two separate time, the runs tested for performance are marked with a *. The other run was used to train the models considered in the next section.

device	run	min	max	mean	std
21B	1	25.4	46.7	34.6	4.6
21B	2*	29.6	51.3	40.8	6.5
23B	2*	26.3	44.1	35.4	4.9
23B	3	22.8	46.6	36.6	6.9
24B	2*	27.1	47.2	36.4	6.2
24B	3	27.3	48.4	36.4	6.5
25B	1	27.6	49.2	38.0	6.6
25B	2*	29.8	57.7	43.8	9.9

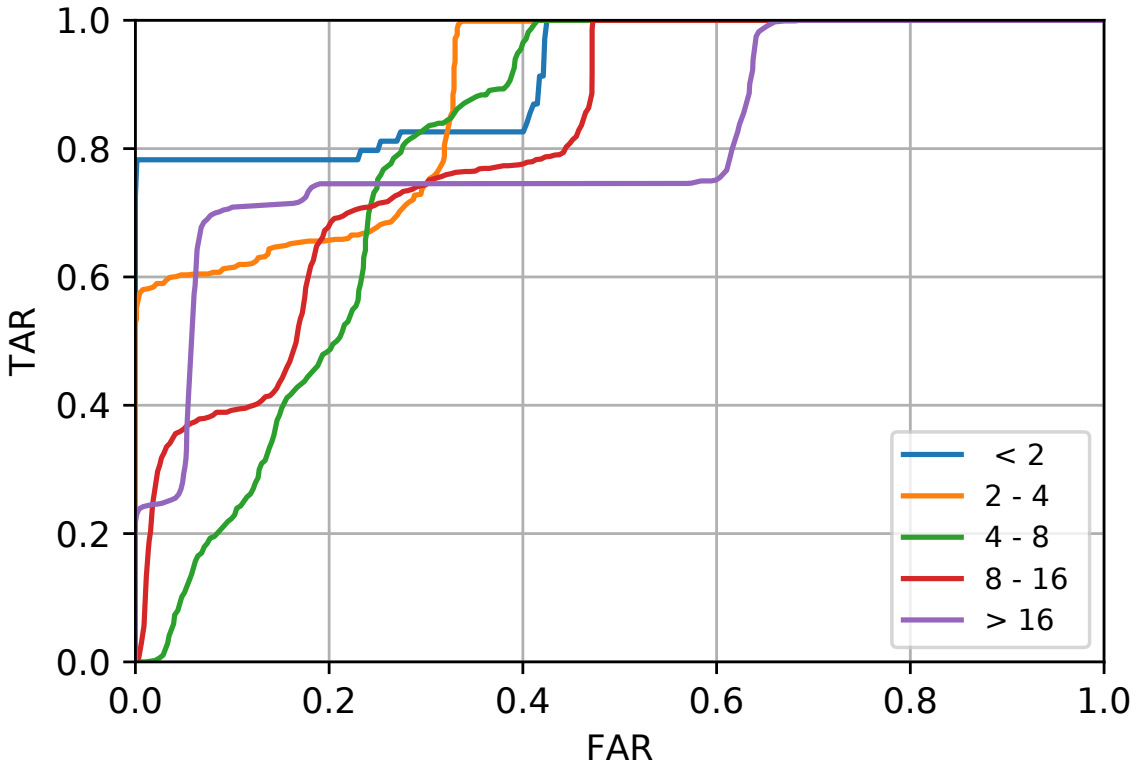


Figure 4.9: Breakdown of performance (in terms of receiver operating curves) with respect to the absolute change in temperature from the average temperature of reference data. Larger temperature changes result in poor performance, and changes above 4 °C to 8 °C show substantial confusion in the fingerprinter.

4.6 Compensating for drift

Two approaches to compensate for drift are considered. First, features that are "permanent" with regards to changes in carrier frequency are considered. Since the drift observed was primarily due to the oscillator these features are largely unaffected changes in temperature. However, their ability to distinguish between devices was significantly less than the original non-permanent features. Next, predicting fingerprinter output as a function of temperature via modeling is considered. This allows the use of features that are not "permanent", but requires some knowledge of the DUT's temperature, with the resulting models being specific to each transmitter.

4.6.1 Features unaffected by CFO

It was thought that the FFT coefficients used as features are independent of the carrier frequency, and consequently would be unaffected by drift in the oscillator. Previous work (in Chapter 2) found that these features can continue to distinguish between devices even when the carrier frequency is changed. However, in Section 4.4 it was shown that the oscillator is primarily responsible for drift. Thus, the oscillator must contribute to the RF features used. Most likely this is due to changes in the carrier frequency offset of the transmitter, but could also be changes in behavior of the DAC or other parts of the RF transciever that depend on the local oscillator rate. We first consider methods to remove any effects of the CFO.

The features found using the FFT use the modulated signal, which includes some information about the carrier, or rather the CFO since the bins chosen are based on the expected carrier frequency. The Hilbert transform was used to find the signal envelope, which should contain no information about the CFO. Frequency based features can then be extracted using the signal envelope [44]. In [43] the Hilbert transform is used to remove any information about the carrier, as it's suggested that an attacker can imitate features based on CFO more easily than other features. It would also be possible to use a receiver with the PLL locked on the center frequency to remove the effects of CFO. This is not practical with the current experimental setup, which uses an oscilloscope to capture records.

The resulting features are very consistent with regards to temperature, shown in Figure 4.10. Features from the four devices tested drift very little with regards to temperature. Although there are some substantial outliers, the majority of records for all devices are within one standard deviation of the expected fingerprinter output for the DUT. The receiver operating curves, shown in Figure 4.11, exhibit only slightly better performance than random guessing. There is not a large difference in performance between most of the temperature ranges. Performance is best for large changes in temperature, covering changes greater than 4 °C. Smaller changes have worse performance, surprisingly. Unfortunately, this shows that features based on the envelope are unable to reliably distinguish between devices. However, this confirms that oscillator drift changing the CFO is the primary cause of observed drift.

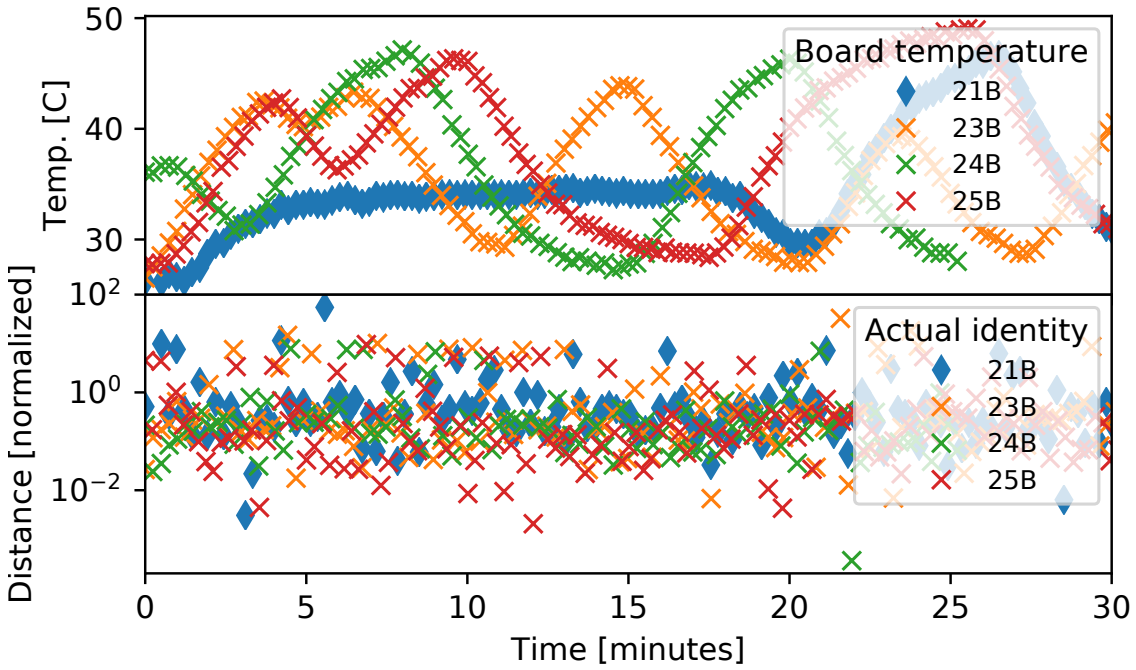


Figure 4.10: Internal temperature of each device as it was heated (upper), with corresponding fingerprinter output using features extracted from the signal envelope (lower). The fingerprinter uses reference data from transmitter 21B. Each device has relatively consistent behavior, in spite of large changes in temperature. However, there is very little distinguishability between devices (although this is one of the worse examples).

Examining the features shows only a small differences in the frequency bins used (see Figure 4.12). This suggests that the RF features used gain most of their distinguishability from the carrier. This is unexpected and could not be seen in the original feature set. The effect of the carrier was very minor – total drift due to temperature is specified at around 4 kHz (2 ppm) over the range of interest [85]. The bin width used in each feature was around 6 kHz. Consequently, the effect was not immediately obvious in the features due to the small shift and wide bin width.

This is probably an issue specific to the type of transmitters used: the devices tested are SDRs, which contain higher end hardware than the consumer wireless cards tested in many previous works. Previous works have used Zigbee transmitters, WiFi cards, or similarly low-cost consumer hardware. It is possible the transmitters used here provides a very consistent baseband signal which provides very little variation for fingerprinting. There are no other works fingerprinting this model of transmitter (the Ettus B210) that we are aware of. The Ettus N210 has been fingerprinted [9].

The dependency of these features on CFO may be a general issue in how many frequency based features are found. In this case it occurred since the modulated data was captured with the oscilloscope. CFO impacts the time domain signal since the modulated data was

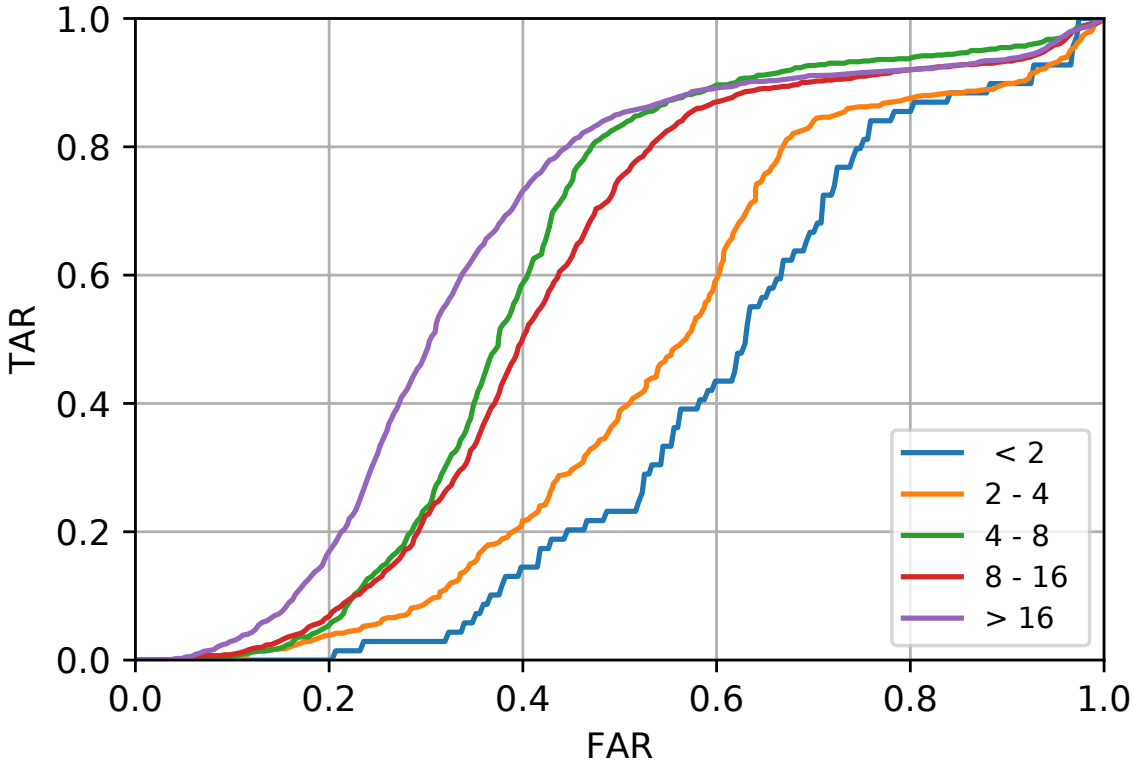


Figure 4.11: Performance breakdown using features extracted from signal envelope (found using the Hilbert transform). Performance was worse for all temperature ranges than the original features in Figure 4.9. However, all records with temperature greater than 4 °C exhibit similar performance.

captured without using a PLL locked to the carrier frequency. Details vary in the literature, but many works use spectrum or signal analyzers [9, 89]. These are tuned to a general frequency to capture a signal, so the signals found are probably also affected by the carrier frequency. None specifically analyze the effects of carrier frequency on these features, so it seems likely that some works finding that the FFT provides good features are actually measuring the carrier indirectly as happened here. In [9] performance of a fingerprinting using a spectrum analyzer is compared with fingerprinting using an SDR. The DUTs consisted of model N200 USRPs, also from Ettus research. It was found that the USRPs have much worse performance, attributed to the lower-end hardware compared to the signal analyzer. However, it is possible the signal analyzer retains traces of the CFO while the USRPs do not. In [43] the features extracted directly from the signal captured by a signal analyzer are compared with those extracted from the envelope found with the Hilbert transform. Using the envelope only decreases performance slightly when identifying Tmote Sky sensor nodes. This suggests that lower cost hardware likely has greater variability in the signal envelop, which is advantageous for fingerprinting.

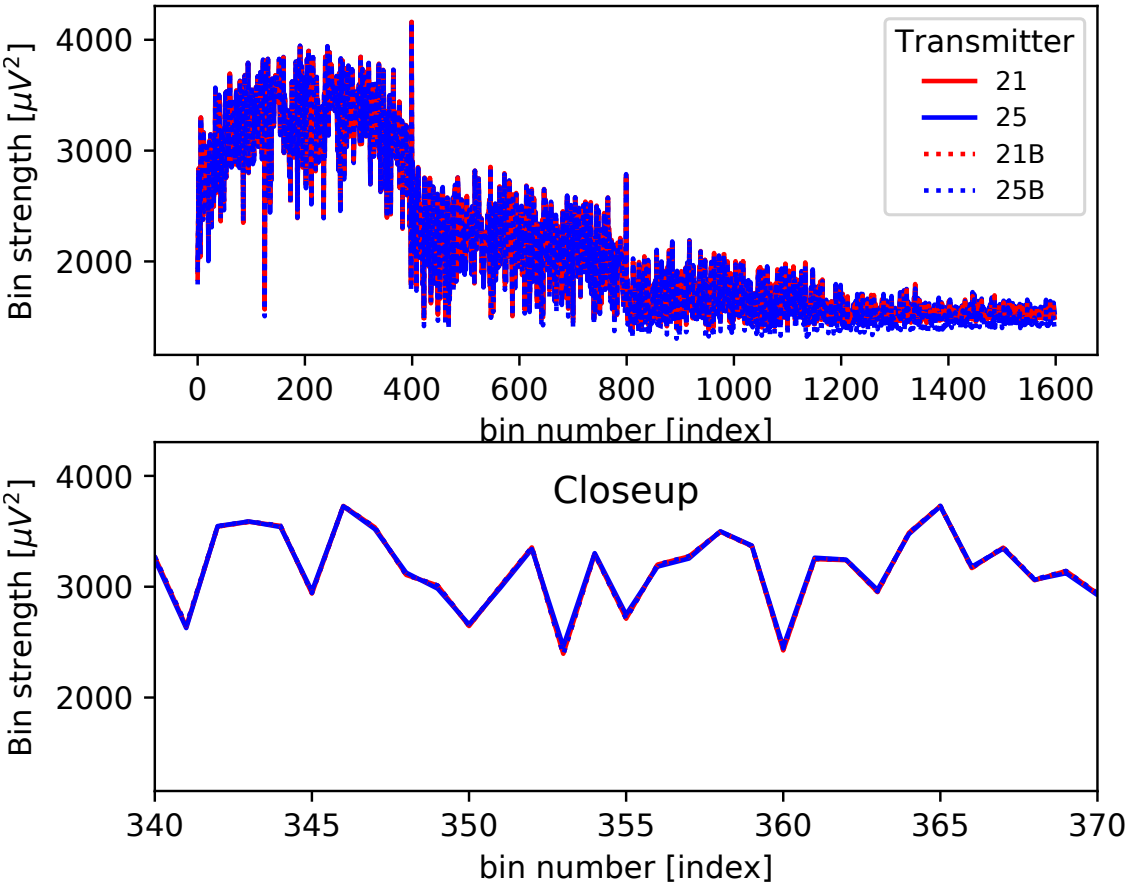


Figure 4.12: RF features, found using Hilbert transform to obtain signal envelope. The features from every device are very similar when the envelope is used.

4.6.2 Modeling drift

When features not subject to drift can't be found some correction must be taken. Here, we develop a model to compensate for verifier output given temperature, and fit to data from each transmitter. The best model would one that can be used to predict variation over an entire class of devices. A device specific model incurs additional overhead during the training process. Additionally, if a device goes through regular temperature cycles (such as day and night) a model can be used to predict future behavior. The number of records used to train an effective model is also considered. This is an import consideration, as collecting data for training introduces some overhead in the system.

In contrast with the tracking methods discussed in Section 4.2.2, modeling methods can work when a device is not observed for a long interval. This may occur when a transmitter is turned off, moves, or stops transmitting. This advantage comes at the cost of requiring an accurate model of device behavior and knowledge of its temperature. In a practical system

a combination of tracking and modeling would be advantageous.

To utilize the models, we assume that accurate temperature information is available for the DUT. How this is measured and reported is outside the scope of this work, but several possibilities are available. Devices self-reporting undermines the utility of fingerprinting as it requires a cooperating user and modifications to authentication protocols to include verified temperature information. It may be possible to infer temperature from other factors (such as in [16] where WiFi devices in same room or building are assumed to have a similar temperature).

Models used

The fingerprinter output for the DUT can be modeled so that $E(T)$ is the expected output for the legitimate device when it is at temperature T. Denoting the verifier output for the DUT as $V(D)$, a new verifier is then found as

$$V_C(D;T) = \text{abs}(V(D) - E(T)) \tag{4.1}$$

That is, for each record the temperature is used to predict the fingerprinter's output value. The absolute difference between the predicted and actual fingerprinter output is taken. This should be close to 0 when the DUT is legitimate and larger when it is an attacker.

Multivariable polynomial regression is used for the models, making use of the following variables:

- T The device temperature, as measured by the sensor on the RF transceiver.
- Δ The derivative of device temperature, found using the difference method.
- I A categorical variable. $I = 1$ when the temperature is increasing, otherwise $I = 0$.

A 100 point moving average of the temperature is used to reduce noise and improve the estimates of the derivative. As discussed earlier, the oscillator exhibits substantial hysteresis. This can be incorporated into the model by using a categorical variable, which allows for different behavior when the device is heated or cooled.

These variables are combined in several models, using a 3rdpolynomial. This was chosen as it incorporates most of the variance in the data, but found to still generalize well to temperatures outside the range of the training data. The models examined were:

- **basic** Fitting temperature only. $E(T) = a + bT + cT^2 + dT^3$
- **split** Only temperature fitted, but including a categorical variable. $E(T) = a + bIT + cIT^2 + dIT^3 + e(I-1)T + f(I-1)T^2 + g(I-1)T^3$
- **splitdelta** The **split** model, plus incorporating the rate of change of temperature. $E(T) = a + h\Delta + i\Delta^2 + j\Delta^3 + bIT + cIT^2 + dIT^3 + e(I-1)T + f(I-1)T^2 + g(I-1)T^3$

Results

To evaluate the models, the data described in Section 4.5 is used. As in that section, the fingerprinter output is evaluated directly, as well as the receiver operating characteristics. The first run of data is used to fit a model for each transmitter. An additional run of data from each transmitter was then used to evaluate the performance of the fingerprinter, using Equation 4.1. The run used for testing has a slightly higher temperature range (by about 5 °C) so this also tests that the models can generalize slightly.

The resulting distances are used to find the receiver operating characteristics (Figure 4.13). The basic model, only taking temperature into account, performs better than without any model for most FARs. It also gives a much more predictable curve, suggesting the results will hold over a range of datasets. The **split** and **splitdelta** models perform best, with no clear difference in performance between them. Clearly, incorporating hysteresis improves the model substantially. The temperature rate of change adds very little information to the model.

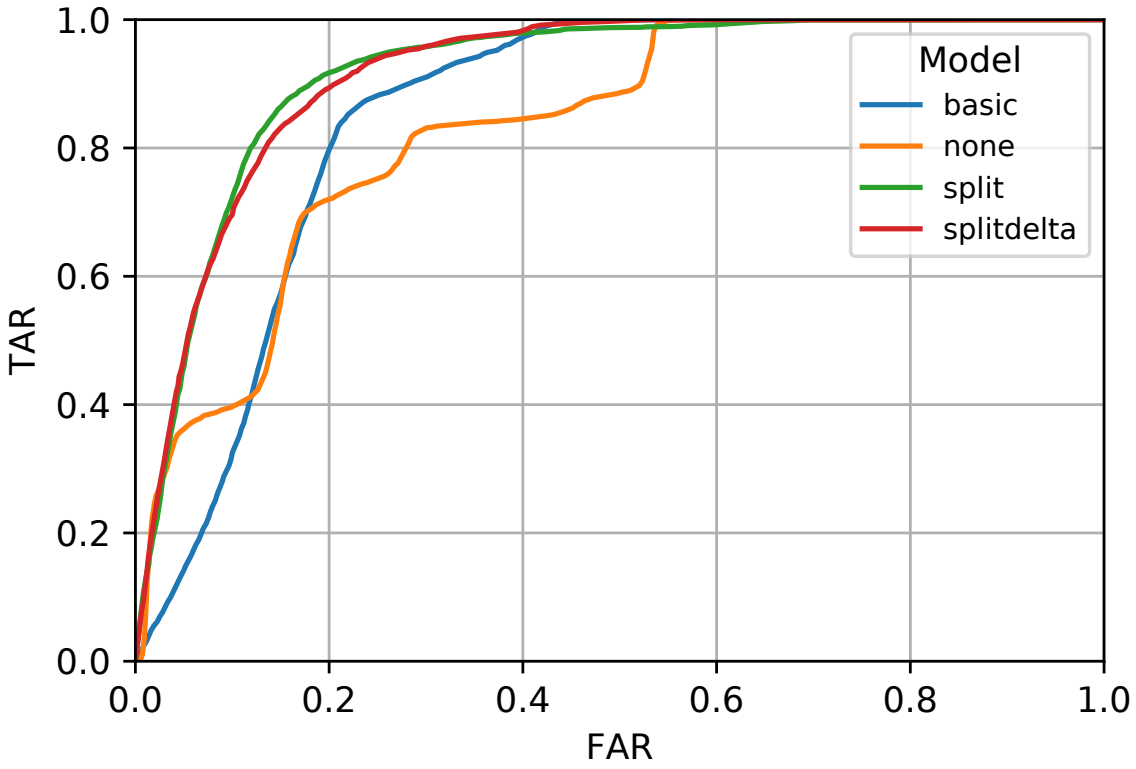


Figure 4.13: Performance of various regression based models. None indicates that the verifier output was used directly. Adding a model improves performance, and incorporating hysteresis (split) further improves it. Incorporating rate of change (delta) does very little.

The predicted distances are shown in Figure 4.14. The distances are small, with the DUT always being within a standard deviation of the reference data. The small distances are

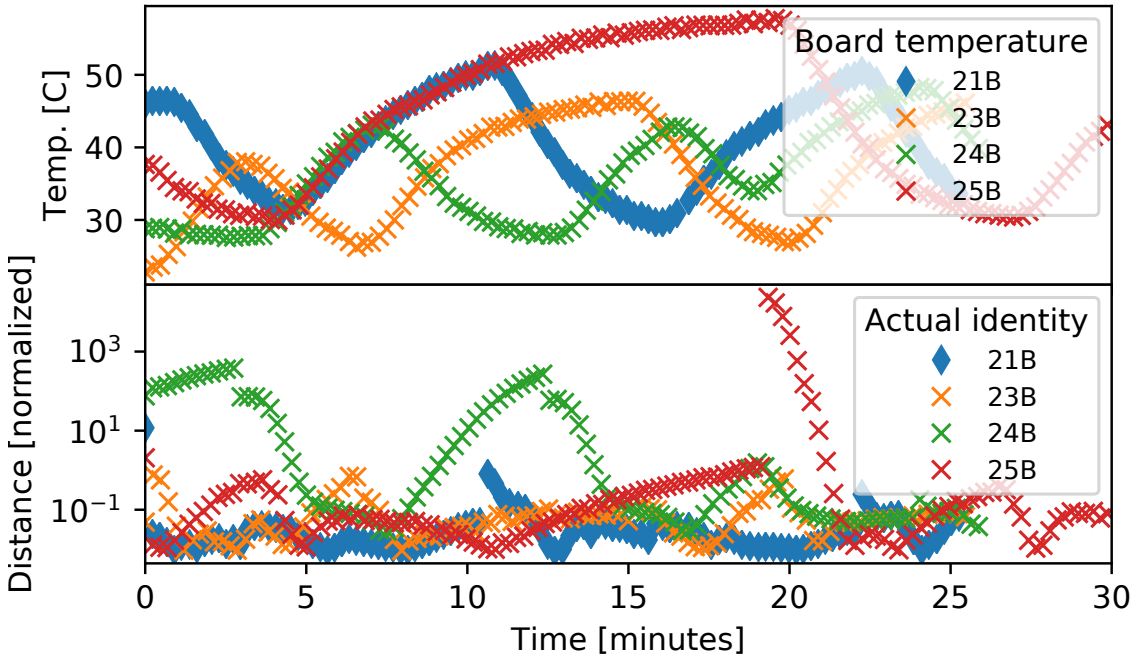


Figure 4.14: Internal temperature of each device as it is heated (upper), with corresponding fingerprinter output distances taking the difference from the model **splitdelta**. Using reference data from transmitter 21B (lower). The distances for the DUT are smallest, except where transitions in temperature occur.

due to the model, and it's use of smoothed temperature data providing a better prediction. However, most of the other devices exhibit small distances for the same reason. What is most important is that in most cases there is separation between the DUT and other devices. This is a substantial improvement over the original distances shown in Figure 4.8. This suggests that each model can generalize somewhat outside the temperature range of the data used to create it, but has trouble when the temperature changes from increasing to decreasing.

The best model would be one that applies to a class of transmitters, and continues to work well over time. In Table 4.2 the coefficients found for model **basic** are shown. Similar coefficients should be found for data collected from the same transmitter at different times, especially for the lower order coefficients. This is the case for most of the transmitters. What variation there is likely due to small changes in airflow in the test setup, rather than changes in an oscillator's actual behavior. Effort was made to place transmitters in the same position, but there was still some variation in the test setup between each run. Unfortunately, the models do not generalize beyond a single transmitter. Enough differences exist between devices to prevent re-using a model for multiple devices.

Using a regression based model helps performance, but has some limitations. The models found work well for the temperature range trained, and can generalize outside it slightly. When switching from increasing to decreasing temperature the model has some difficulties.

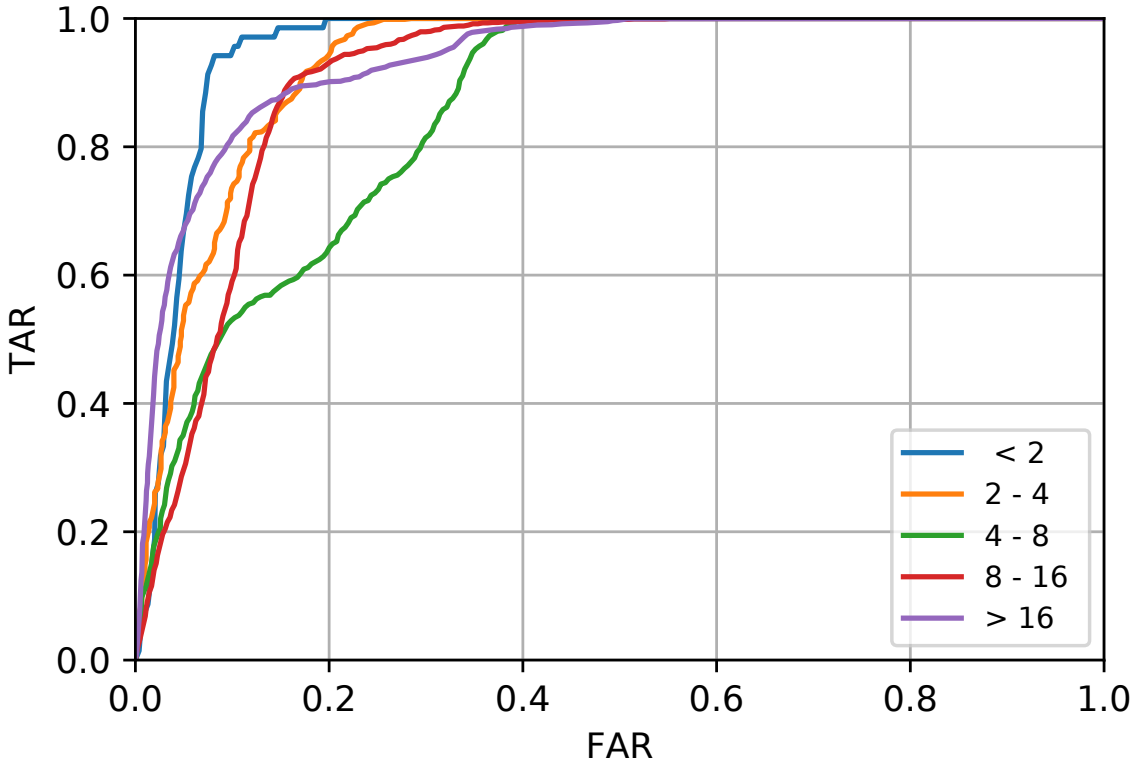


Figure 4.15: Breakdown of performance by temperature for model splitdelta. It can be seen that temperature has a much smaller impact on performance, and the receiver operating curves are much more predictable than in Figure 4.9.

Additionally, the verifier output for legitimate and some other devices will always overlap for some range of temperatures, causing poor performance. A combination of predicting features, using the predicted verifier output in concert with other features, or tracking methods would help avoid this. Tracking can compensate for immediate changes, while modeling can be used to forecast future behavior and examine cyclic changes such as day and night temperature cycles.

4.6.3 Amount of data to fit model

The last consideration is the number of records required to accurately model the temperature dependent behavior of a transmitter. Obtaining reference data from a legitimate device can be difficult. Modeling is beneficial if it can reduce the amount of reference data needed when a device operates at a new temperature.

To test the number of records needed to model behavior, the model is fit with a progressively smaller set of data and the mean squared error is measured. The records used to fit the model are randomly chosen out of about 1500 possible records. The mean and standard deviation

Table 4.2: Coefficients found for the basic model, for each transmitter. Two runs of data shown for each transmitter, for the most part the coefficients for runs from the same transmitter are much closer than runs from different transmitters. This suggests that the models found will continue to work for the same device, but cannot be generalized across devices.

TX	run	a	b	c	d
21B	1	0.1828	0.2433	-0.0068	-0.0005
21B	2	0.1805	0.0648	-0.0044	0.0003
23B	2	0.0383	0.0632	-0.0017	-0.0002
23B	3	0.1015	0.0641	-0.0023	-0.0002
24B	2	-0.0695	-0.0982	0.0011	0.0007
24B	3	-0.1614	-0.1104	0.0039	0.0003
25B	1	0.1155	-0.0127	-0.0028	0.0005
25B	2	-0.0424	0.0306	0.0005	-0.0000

Table 4.3: Performance, in terms of the mean squared error of the test data, of the **splitdelta** model fit with varying amounts of data. Mean performance plus or minus one standard deviation shown for 100 runs. The mean performance improves up to 512 records, and after this the standard deviation continues to improve.

	64	128	256	512	1024
21B	0.56 ± 1.98	0.09 ± 0.05	0.07 ± 0.02	0.06 ± 0.01	0.06 ± 0.00
23B	0.60 ± 1.32	0.12 ± 0.16	0.07 ± 0.02	0.07 ± 0.01	0.06 ± 0.00
24B	0.47 ± 0.94	0.12 ± 0.15	0.07 ± 0.02	0.06 ± 0.01	0.06 ± 0.00
25B	0.41 ± 0.88	0.14 ± 0.23	0.08 ± 0.03	0.07 ± 0.01	0.06 ± 0.01

of the mean squared error is shown in Table 4.3. The mean performance doesn't improve substantially with more than 512 records, although the standard deviation continues to improve. This suggests the models can be accurately found with fewer records than this. In contrast, around 1500 records are used as reference data for the verifier in the results previously presented. Multiple sets of reference data would be required for a verifier to cover a large temperature range, whereas the model can provide good performance over the entire range with less records than would be required for a single verifier.

Another aspect of the amount of data required to fit these models is the temperature range to be observed, how many samples are required, and which points in the temperature range must be sampled. Unfortunately, as has been demonstrated, the behavior of each device is different so these points cannot be determined for a device without observing it's behavior. For some devices, such as transmitter 21, there are abrupt changes in behavior – consequently these changes must be included for an accurate model. These changes may correspond with the turnover point in the oscillator's crystal. For these devices, a model will not generalize to a large range of temperatures if the change point is not included in the training data. Some transmitters exhibit much simpler behavior which can be fit with a few points, and will likely generalize outside that. The models used are based on a third

order polynomial, so if exact measurements were had only four points would be required to find the model. However, substantial noise is present in temperature measurements so more observations are required. The exact number will depend on the amount of noise present, and how closely an individual transmitter's behavior matches the model. Additionally, when a transmitter exhibits hysteresis the number of observations required doubles as examples of both increasing and decreasing temperature are required.

4.7 Conclusions

This examination of temperature based drift in fingerprints has found a number of notable results. First, the oscillator was determined to be the primary cause of drift in the devices examined. The other components were found to contribute little or nothing. The oscillator in the devices used is a quartz crystal oscillator, similar to those found in most wireless devices. It is expected that hardware using quartz oscillators will exhibit similar drift.

Second, it was found that the RF features used are very dependent on the behavior of the oscillator and, consequently, quite susceptible to drift. This is likely an issue with many RF features used in the literature. Fingerprinting features based on the envelope rather than the modulated signal should be used in practice, when possible, to provide resistance to drift. This also shows that temperature is an important experimental variable that should be measure or controlled. In the experiments, the two separate frontends of an Ettus B210 transmitters appear quite different until temperature is controlled. This suggests that a fingerprinting system with devices which experiences temperature changes may exhibit substantially different behavior than testing would suggest.

Lastly, it was demonstrated that models of a transmitters drift can be used to improve performance when drift is present. These models were found to be specific to each device, but still allow for improved performance with a limited set of reference data.

There are a number of future research areas that may be useful. It is expected that oscillator behavior will be similar in wireless cards and other low cost devices. However, it has been shown that features based on the envelope can distinguish between these devices effectively, unlike the USRPs used in this work. It would be good to examine these devices to see how features based on the envelope are affected by drift. Similarly, examining transmitters with a more powerful amplifier may show additional drift. Investigating additional approaches to compensate for drift could be beneficial. When modeling the fingerprinter's output there are unavoidable situations where the fingerprinter output for a legitimate device overlaps with an impostor. These are more common as large changes in temperature occur. Approaches based on features rather than verifier output would make this less likely. Likewise, a combination of tracking and modeling would be useful. Similarly, since drift is based on oscillator behavior it may be possible to identify devices based on the rate of drift. Observations over a substantial period of time, such as a day and night temperature cycle, may provide an accurate enough

model of a specific device's drift to identify it.

Chapter 5

Conclusions

Several extensions to current fingerprinting methods have been demonstrated. It has been shown that fingerprinting methods can be applied to transmitters which change bandwidth, modulation, and carrier frequency. Crowdsourcing methods have been examined, showing several possibilities for combining fingerprinting measurements. Finally, it has been shown that fingerprinting performance suffers when transmitters experience substantial drift from changes in temperature, but this can be compensated for. These open up some possibilities for future examination.

Some features are heavily impacted by changes in transmitter configuration, discussed in Chapter 2. This will become a significant issue as cognitive radios become more widespread. A method to transform features to be invariant with respect to several types of configuration was proposed and demonstrated. With it, transmitters can be re-identified after changes in modulation or bandwidth. This was demonstrated using RF frequency features based on the FFT. Chapter 4 suggests that the majority of variation in these features is primarily due to each transmitter's CFO, and that the majority of variation in the features used is a proxy for variation in the carrier frequency of each device. In this case, it would be simpler to use CFO directly to re-identify transmitters changing transmission parameters. This would confirm the earlier hypothesis in Section 2.1.1 that predefined features (CFO in this case) are less susceptible to configuration dependency. It would be good to test CFO and other predefined features to see how they are impacted by changes in modulation, bandwidth, and other configuration.

In Chapter 3 several approaches to crowdsourcing fingerprinting data were examined. Combinations of features, fingerprinter output, and raw observations of the signal were considered. Although the best method was found to depend heavily on the feature set used, combinations of features performed well in all situations tested. It remains to be seen if these techniques will work with the constraints of a practical network. In this case, the effects of the channel would need to be analyzed further, including how channel equalization may affect fingerprints. Further analysis including determining which feature types are better suited to a

particular combination type would also be useful.

In Chapter 4 it was demonstrated that the oscillator is the primary cause of feature drift. This was concluded by selectively heating various components on the board, and observing fingerprinter output. This same technique could be extended to determine help determine which components contribute to the variation in fingerprints, for components that are sensitive to temperature changes. The changes in fingerprinter output due to temperature can be compensated for by extracting features from the signal envelope rather than raw observations. However, this decreases performance substantially. Models of a transmitter's behavior based on temperature were found, which removed the majority of the effects of drift due to temperature. For further applicability these results should be tested on a larger range of devices, particularly lower cost transmitters. Additionally, investigating clustering methods or other methods to track changes in features may provide better performance, as some limits were found on the performance of predicting fingerprinter output. Modeling features rather than fingerprinter output would also enhance the performance in many situations, although it is unknown how well some types of features can be modeled. Testing a wider variety of features would also be beneficial. Modeling predefined features may give a better indication of how the signal is changing and provide further insights. It was also found that the drift experienced by each device is unique. This could be used over longer periods of time to identify a device based on drift over time.

Several extensions to fingerprinting have been proposed, which will allow a greater variety of applications. Future research directions have been suggested, which may improve the methods proposed. A number of other areas of fingerprinting are also of interest. Some attacks have been developed against fingerprinting systems, but have been limited in their applications, and have a weak attacker model. Also of interest are theoretical bounds on the capabilities of fingerprinting. Some steps have been made in this direction, but a complete theoretical basis is not available for fingerprinting.

Impersonation and replay attacks against fingerprinting systems have only been superficially examined since they were originally proposed (see Section 1.3.2. While it has been shown that predefined features such as CFO and IQ imbalance are susceptible to attack other features such as those based on the FFT have not been successfully impersonate. It was shown in Chapter 4 that in several cases RF features incorporate information about the carrier frequency. It is likely that, in such cases, these features are more vulnerable to an attacker than previously thought. Additionally, the attackers analyzed in the literature have been very naive. An attack model incorporating a more sophisticated attacker, such as one using calibrated hardware and taking steps to remove noise from observed signals, may have a greater success rate against features currently thought to be resistant to impersonation attacks.

Several limited models have been developed around fingerprinting, covered in Chapter 1, including models of how some components contribute to fingerprints, and measures of the number of devices that can be reliably identified at a set accuracy level (known as user capac-

ity). User capacity will become more important as applications of fingerprinting involving larger numbers of devices are made. As more low cost transmitters become available, it will be useful to put specific bounds on how likely it is that an attacker can find a similar device. Understanding the sources of fingerprinting and having a model of how all transmitter components may contribute to features will also help with this.

Wireless fingerprinting has found a number of uses. The extensions proposed here will allow a number of new applications, including identifying devices which change configuration, crowdsourcing measurements, and tracking drift in features due to changes in temperature.

Bibliography

[1] S. Andrews, R. M. Gerdes, and M. Li, "Crowdsourced measurements for device fingerprinting," in *Proc. of the 12th Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*. ACM, 2019, pp. 72–82.

[2] B. Danev, D. Zanetti, and S. Capkun, "Types and origins of fingerprints," in *Digital Fingerprinting*, C. Wang, R. M. Gerdes, Y. Guan, and S. K. Kasera, Eds. Springer, 2016, ch. 1, pp. 5–29.

[3] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 94–104, 2016.

[4] G. Baldini and G. Steri, "A survey of techniques for the identification of mobile phones using the physical fingerprints of the built-in components," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1761–1789, third quarter 2017.

[5] K. I Talbot, P. R Duley, and M. H Hyatt, "Specific emitter identification and verification," *Technology Review Journal*, 01 2003.

[6] L. E. Langley, "Specific emitter identification (SEI) and classical parameter fusion technology," in *Proceedings of WESCON'93*. IEEE, 1993, pp. 377–381.

[7] H. C. Choe, C. E. Poole, M. Y. Andrea, and H. H. Szu, "Novel identification of intercepted signals from unknown radio transmitters," in *Wavelet Applications II*, vol. 2491. International Society for Optics and Photonics, 1995, pp. 504–517.

[8] R. Yu, Y. Zhang, Y. Liu, S. Gjessing, and M. Guizani, "Securing cognitive radio networks against primary user emulation attacks," *IEEE Network*, vol. 29, no. 4, pp. 68–74, 2015.

[9] S. U. Rehman, K. W. Sowerby, and C. Coghill, "Radio-frequency fingerprinting for mitigating primary user emulation attack in low-end cognitive radios," *IET Communications*, vol. 8, no. 8, pp. 1274–1284, 2014.

[10] Z. Zhuang, X. Ji, T. Zhang, J. Zhang, W. Xu, Z. Li, and Y. Liu, "FBSleuth: Fake base station forensics via radio frequency fingerprinting," in *Proc. of the 2018 ACM Asia Conference on Computer and Communications Security*, 2018, pp. 261–272.

[11] A. C. Polak, S. Dolatshahi, and D. L. Goeckel, "Identifying wireless users via transmitter imperfections," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 7, pp. 1469–1479, 2011.

[12] K. B. Rasmussen and S. Capkun, "Implications of radio fingerprinting on the security of sensor networks," in *Third International Conference on Security and Privacy in Communications Networks (SecureComm)*. IEEE, 2007, pp. 331–340.

[13] P. Scanlon, I. O. Kennedy, and Y. Liu, "Feature extraction approaches to RF fingerprinting for device identification in femtocells," *Bell Labs Technical Journal*, vol. 15, no. 3, pp. 141–151, 2010.

[14] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. Moura, "A deep learning approach to IoT authentication," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[15] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, "Identifying ECUs using inimitable characteristics of signals in controller area networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4757–4770, 2018.

[16] F. Lanze, A. Panchenko, B. Braatz, and T. Engel, "Letting the puss in boots sweat: Detecting fake access points using dependency of clock skews on temperature," in *Proc. of the 9th ACM symposium on Information, Computer and Communications Security*. ACM, 2014, pp. 3–14.

[17] J. Hall, M. Barbeau, and E. Kranakis, "Enhancing intrusion detection in wireless networks using radio frequency fingerprinting." in *Communications, internet, and information technology*, 2004, pp. 201–206.

[18] B. Danev, S. Capkun, R. Jayaram Masti, and T. S. Benjamin, "Towards practical identification of HF RFID devices," *ACM transactions on Information and System Security (TISSEC)*, vol. 15, no. 2, p. 7, 2012.

[19] L. Yang, P. Peng, F. Dang, C. Wang, X. Y. Li, and Y. Liu, "Anti-counterfeiting via federated RFID tags' fingerprints and geometric relationships," in *IEEE Conference on Computer Communications (INFOCOM)*, 2015.

[20] R. M. Gerdes and S. Mallick, "Physical-layer detection of hardware keyloggers," in *International Symposium on Recent Advances in Intrusion Detection*. Springer, 2015, pp. 26–47.

[21] C. Yang and A. P. Sample, "EM-ID: Tag-less identification of electrical devices via electromagnetic emissions," in *2016 IEEE International Conference on RFID*. IEEE, 2016, pp. 1–8.

[22] P. Eckersley, "How unique is your web browser?" in *International Symposium on Privacy Enhancing Technologies*. Springer, 2010, pp. 1–18.

[23] Z. Ba, S. Piao, X. Fu, D. Koutsonikolas, A. Mohaisen, and K. Ren, "ABC: Enabling smartphone authentication with built-in camera," in *25th Annual Network and Distributed System Security Symposium, NDSS 2018*, 2018.

[24] J. Zhang, A. Beresford, and I. Sheret, "SensorID: Sensor calibration fingerprinting for smartphones." IEEE, 2019.

[25] B. Perez, M. Musolesi, and G. Stringhini, "Fatal attraction: Identifying mobile devices through electromagnetic emissions," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks(WiSec)*. ACM, 2019, pp. 163–173.

[26] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels," *arXiv preprint arXiv:0907.4919*, 2009.

[27] S. Fang, Y. Liu, W. Shen, and H. Zhu, "Where are you from? Confusing location distinction using virtual multipath camouflage," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 225–236.

[28] V. Kumar, J.-M. Park, and K. Bian, "Blind transmitter authentication for spectrum security and enforcement," in *Proc. of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 787–798.

[29] L. Yang, Z. Zhang, B. Y. Zhao, C. Kruegel, and H. Zheng, "Enforcing dynamic spectrum access with spectrum permits," in *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, 2012, pp. 195–204.

[30] X. Jin, J. Sun, R. Zhang, and Y. Zhang, "SafeDSA: Safeguard dynamic spectrum access against fake secondary users," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 304–315.

[31] W. Wang, Z. Sun, K. Ren, and B. Zhu, "User capacity of wireless physical-layer identification," *IEEE Access*, vol. 5, pp. 3353–3368, 2017.

[32] L. Wong, W. Headley, S. Andrews, R. Gerdes, and A. Michaels, "Clustering learned CNN features from raw I/Q data for emitter identification," in *IEEE Military Communications Conference (MILCOM)*, 2018.

[33] K. Youssef, L. Bouchard, K. Haigh, J. Silovsky, B. Thapa, and C. Vander Valk, "Machine learning approach to RF transmitter identification," *IEEE Journal of Radio Frequency Identification*, vol. 2, no. 4, pp. 197–205, 2018.

[34] A. C. Polak and D. L. Goeckel, "Rf fingerprinting of users who actively mask their identities with artificial distortion," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. IEEE, 2011, pp. 270–274.

[35] M. Edman and B. Yener, "Active attacks against modulation-based radiometric identification," *Rensselaer Institute of Technology, Technical report*, pp. 09–02, 2009.

[36] B. Danev, H. Luecken, S. Capkun, and K. El Defrawy, "Attacks on physical-layer identification," in *Proceedings of the third ACM conference on Wireless network security*. ACM, 2010, pp. 89–98.

[37] R. M. Gerdes, M. Mina, and T. E. Daniels, "Towards a framework for evaluating the security of physical-layer identification systems," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2012, pp. 328–348.

[38] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 2008, pp. 116–127.

[39] T. J. Bihl, K. W. Bauer, M. A. Temple, and B. Ramsey, "Dimensional reduction analysis for physical layer device fingerprints with application to ZigBee and Z-Wave devices," in *MILCOM 2015-2015 IEEE Military Communications Conference*. IEEE, 2015, pp. 360–365.

[40] J. Hua, H. Sun, Z. Shen, Z. Qian, and S. Zhong, "Accurate and efficient wireless device fingerprinting using channel state information," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1700–1708.

[41] P. Liu, P. Yang, W.-Z. Song, Y. Yan, and X.-Y. Li, "Real-time identification of rogue wifi connections using environment-independent physical features," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 190–198.

[42] P. K. Harmer, D. R. Reising, and M. A. Temple, "Classifier selection for physical layer security augmentation in cognitive radio networks," in *IEEE International Conference on Communications (ICC)*, 2013, pp. 2846–2851.

[43] B. Danev and S. Capkun, "Transient-based identification of wireless sensor nodes," in *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*. IEEE Computer Society, 2009, pp. 25–36.

[44] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 160–167, 2018.

[45] W. Wang, Z. Sun, S. Piao, B. Zhu, and K. Ren, "Wireless physical-layer identification: Modeling and validation," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2091–2106, 2016.

[46] A. M. Salama, M. Li, and D. Yang, "Optimal crowdsourced channel monitoring in cognitive radio networks," in *GLOBECOM*. IEEE, 2017, pp. 1–6.

[47] A. Dutta and M. Chiang, "'See something, say something' crowdsourced enforcement of spectrum policies," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 67–80, 2016.

[48] M. Pospíšil, R. Maršálek, and T. Gotthans, "Wireless device classification through transmitter imperfections – evaluation of performance degradation due to the chip heating," in *2017 IEEE Radio and Wireless Symposium (RWS)*. IEEE, 2017, pp. 169–172.

[49] S. Andrews, R. M. Gerdes, and M. Li, "Towards physical layer identification of cognitive radio devices," in *Conference on Communications and Network Security (CNS)*. IEEE, 2017.

[50] E. Hossain, D. Niyato, and Z. Han, *Dynamic spectrum access and management in cognitive radio networks*. Cambridge Univ. Press, 2009.

[51] A. C. Polak and D. L. Goeckel, "RF fingerprinting of users who actively mask their identities with artificial distortion," in *Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. IEEE, 2011, pp. 270–274.

[52] R. M. Gerdes, M. Mina, S. F. Russell, and T. E. Daniels, "Physical-layer identification of wired Ethernet devices," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1339–1353, 2012.

[53] Y. Sharaf-Dabbagh and W. Saad, "Transfer learning for device fingerprinting with application to cognitive radio networks," in *26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2015, pp. 2138–2142.

[54] T. C. Clancy and N. Goergen, "Security in cognitive radio networks: Threats and mitigation," in *Conference on Cognitive Radio Oriented Wireless Networks and Communications*. IEEE, 2008, pp. 1–8.

[55] Tektronix, Inc., "DPO7000 series datasheet," 2017, datasheet.

[56] Ettus Research, "USRP N210 datasheet," 2012, datasheet.

[57] ——, "USRP B200/B210 specification sheet," 2017, datasheet.

[58] R. Bolle, *Guide to Biometrics*. Springer, 2004.

[59] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "SpecGuard: Spectrum misuse detection in dynamic spectrum access systems," in *IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 172–180.

[60] J. Xiong and K. Jamieson, "SecureArray: Improving WiFi security with fine-grained physical-layer information," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 441–452.

[61] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *IEEE Symposium on Security and Privacy (SP)*, 2010, pp. 286–301.

[62] U. Satija, N. Trivedi, G. Biswal, and B. Ramkumar, "Specific emitter identification based on variational mode decomposition and spectral features in single hop and relaying scenarios," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 581–591, 2019.

[63] K. Kim, C. M. Spooner, I. Akbar, and J. H. Reed, "Specific emitter identification for cognitive radio with application to IEEE 802.11," in *Global Telecommunications Conference, (GLOBECOM)*. IEEE, 2008, pp. 1–5.

[64] S. J. Pan, J. T. Kwok, Q. Yang, and J. J. Pan, "Adaptive localization in a dynamic WiFi environment through multi-view learning," in *AAAI*, 2007, pp. 1108–1113.

[65] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. 18th International Conference on Mobile Computing and Networking*. ACM, 2012, pp. 173–184.

[66] R. Zhang, J. Zhang, Y. Zhang, and C. Zhang, "Secure crowdsourcing-based cooperative pectrum sensing," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2526–2534.

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

[68] M. Li, D. Yang, J. Lin, and J. Tang, "Specwatch: A framework for adversarial spectrum monitoring with unknown statistics," *Computer Networks*, vol. 143, pp. 176–190, 2018.

[69] V. Kumar, H. Li, J.-M. J. Park, and K. Bian, "Enforcement in spectrum sharing: Crowdsourced blind authentication of co-channel transmitters," in *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2018, pp. 1–10.

[70] S. U. Rehman, K. W. Sowerby, and C. Coghill, "Radio-frequency fingerprinting for mitigating primary user emulation attack in low-end cognitive radios," *IET Communications*, vol. 8, no. 8, pp. 1274–1284, 2014.

[71] A. Candore, O. Kocabas, and F. Koushanfar, "Robust stable radiometric fingerprinting for wireless devices," in *International Workshop on Hardware-Oriented Security and Trust (HOST)*. IEEE, 2009, pp. 43–49.

[72] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Comm. Surveys Tut.*, vol. 11, no. 1, pp. 116–130, 2009.

[73] S. Andrews, R. M. Gerdes, and M. Li, "Towards physical layer identification of cognitive radio devices," in *Conference on Communications and Network Security (CNS)*. IEEE, 2017.

[74] H. G. Feichtinger, K. Gröchenig, and T. Strohmer, "Efficient numerical methods in non-uniform sampling theory," *Numerische Mathematik*, vol. 69, no. 4, pp. 423–440, 1995.

[75] H. G. Feichtinger and K. Gröchenig, "Theory and practice of irregular sampling," in *Wavelets: mathematics and applications*, 1994, pp. 305–363.

[76] R. G. Vaughan, N. L. Scott, and D. R. White, "The theory of bandpass sampling," *IEEE Transactions on Signal Processing*, vol. 39, no. 9, pp. 1973–1984, 1991.

[77] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[78] N. Kurosawa, H. Kobayashi, K. Maruyama, H. Sugawara, and K. Kobayashi, "Explicit analysis of channel mismatch effects in time-interleaved ADC systems," *IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications*, vol. 48, no. 3, pp. 261–271, 2001.

[79] R. van Otten, "Timing correction of time-interleaved ADCs," Master's thesis, Eindhoven University of Technology, Eindhoven, Netherlands, 2009.

[80] J. Yen, "On nonuniform sampling of bandwidth-limited signals," *IRE Transactions on circuit theory*, vol. 3, no. 4, pp. 251–257, 1956.

[81] S. Park, W. Hao, and C. S. Leung, "Reconstruction of uniformly sampled sequence from nonuniformly sampled transient sequence using symmetric extension," *IEEE Transactions on Signal Process.*, vol. 60, no. 3, pp. 1498–1501, 2012.

[82] J. A. Fessler and B. P. Sutton, "Nonuniform fast Fourier transforms using min-max interpolation," *IEEE Transactions on Signal Process.*, vol. 51, no. 2, pp. 560–574, 2003.

[83] A. C. Polak and D. L. Goeckel, "Wireless device identification based on RF oscillator imperfections," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2492–2501, Dec 2015.

[84] S. Taheri, "Evaluation of Tracking Regimes for, and Security of, PLI Systems," Master's thesis, Utah State University, 2015. [Online]. Available: https://digitalcommons.usu.edu/etd/4549

[85] C. E. Components, "Model 525 temperature compensated crystal oscillator," UNKNOWN, datasheet, Document No. 008-0334-0, Rev. B.

[86] F. L. Walls, "Environmental sensitivities of quartz crystal oscillators," in *Proc. 22nd Annual Precise Time and Time Interval (PTTI) Applications and Planning Meeting*, 1990, pp. 465–477.

[87] "Ettus knowledge base: B200/B210/B200mini/B205mini," https://kb.ettus.com/B200/B210/B200mini/B205mini, accessed: 2019-08-14.

[88] Analog Devices, "RF agile transceiver AD9361, Rev. F," 2017, datasheet.

[89] I. O. Kennedy, M. M. Buddhikot, and K. E. Nolan, "Radio transmitter fingerprinting: A steady state frequency domain approach," in *2008 IEEE 68th Vehicular Technology Conference*, Sep. 2008, pp. 1–5.

[90] T. J. Bihl, K. W. Bauer, and M. A. Temple, "Feature selection for RF fingerprinting with multiple discriminant analysis and using ZigBee device emissions," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1862–1874, 2016.

[91] Texas Instruments, "LM35 precision centigrade temperature sensors," 2017, datasheet.