Active Attacks Against Modulation-based Radiometric Identification

RPI Department of Computer Science Technical Report 09-02

Matthew Edman and Bülent Yener

Rensselaer Polytechnic Institute
Department of Computer Science
Troy, NY 12180 USA
{edmanm2,yener}@cs.rpi.edu

Abstract. Radiometric identification is a recently coined term that describes a broad category of techniques for determining the identity of a wireless device based on unique characteristics of its transmitted signal that result from imperfections and variances in the device's manufacturing processes. Existing techniques are based on extracting and classifying features from either the transient portion of a signal or, most recently, from patterns of modulation errors in a received signal, such as symbol phase and magnitude errors. While the latter approach was shown to be extremely successful in correctly identifying wireless devices using an expensive high-end signal analyzer, its accuracy has not been considered or evaluated under realistic deployment scenarios in the presence of an adversary who actively tries to manipulate his own radiometric signature. Using a software-defined radio platform and an implementation of the IEEE 802.11b PHY layer, we provide preliminary results that suggest a modulation-based radiometric identification system is both feasible and reasonably reliable on commodity hardware. We also experimentally evaluate the effectiveness of an attacker who actively tries to manipulate his radiometric signature in order to impersonate another 802.11b wireless device. We show that even a moderately sophisticated adversary can likely significantly reduce the accuracy of a modulation-based radiometric identification scheme based on a commodity RF hardware platform.

1.1 Introduction

Secure authentication is a difficult problem in general. Secure authentication in wireless ad hoc and sensor networks is potentially even more challenging due to the often unstructured and potentially hostile deployment environments. The nature of a wireless network means that an adversary can easily capture and replay any transmitted information, or insert his own transmitter into the network. Wireless devices in an adversarial environment may even be subject to physical compromise, meaning any unprotected keying material stored on the device that is used for authentication (e.g., as in a PKI-based scheme) may become compromised via physical attacks.

In light of these difficulties, researchers have started looking towards supplementing traditional authentication protocols in a wireless network with biometric-like approaches. Much like fingerprints or iris scans can be used to identify humans, various related techniques have been proposed that attempt to extract a "fingerprint" that can be used to characterize and potentially uniquely identify a physical device in a wireless network. The accuracy of the existing techniques have ranged from software driver-level fingerprinting to identification of a single hardware wireless transceiver. The latter approach, which has recently been termed *radiometric identification* [3], determines the identity of a wireless device based on unique characteristics of its transmitted signal that result from imperfections and variances in the device's manufacturing processes.

The underlying assumption in security systems based on physical layer wireless fingerprinting is that the unique characteristics of a hardware transceiver cannot practicably be replicated or copied from one device to another without noticeably disturbing the fingerprint. Yet, surprisingly, very little attention has been given to the actual performance and reliability of such identification schemes in the presence of a "real world" active adversary. It has simply generally been assumed that it is impossible—or at least prohibitively expensive—to accurately replicate another device's physical fingerprint. While precisely duplicating a wireless transceiver's fingerprint (or *radiometric signature*) is indeed unlikely to be feasible, the proposed biometric-like approaches are inherently inexact and rely only on probabilistic fingerprint matching. As a result, it is not strictly necessary to precisely duplicate a radiometric signature; it just has to be "close enough" some percentage of the time.

While constructing an actual hardware-based wireless transceiver and attempting to tweak it in order to replicate another device's signature would indeed be time consuming and potentially ineffective, there is fortunately an alternate approach. Software-defined radio has recently been emerging as a promising and economical approach to building flexible radio systems that can be adapted to a wide variety of purposes. A software-defined radio system typically couples a hardware RF front end with software running on a host PC that performs signal processing on a signal received by the RF front end. The host PC can also send a baseband signal to the RF front end, which will then upconvert it to RF and transmit the modulated waveform.

The ability to define the transmitted signal in software means the system can be modified for a variety of purposes very quickly. The downside of software-defined radio systems, however, is that they can often have a diminished signal processing capacity compared to a special-purpose hardware system since they are limited by the processing capability of the host PC. As a result, they must often operate at a significantly lower sampling frequency than the special-purpose, expensive signal analyzers used in previous work on radiometric identification [3, 7]. Additionally, it is possible for the less expensive components to be of lower quality and therefore introduce additional noise in the ADC or DAC conversion. Still, we feel that the off-the-shelf software-defined radio platform we use in our work is representative of a real-world radiometric identification deployment scenario, where it is likely infeasible (and unwise) to deploy an expensive special-purpose vector signal analyzer in an unsecured and potentially hostile physical environment.

In this technical report, we begin to evaluate the limits of practical radiometric identification using a software-defined radio platform and an implementation of the IEEE 802.11b PHY layer. We experimentally evaluate the effectiveness of an attacker who actively tries to manipulate his radiometric signature in order to impersonate another 802.11b wireless device. We provide results that suggest even a moderately sophisticated adversary can significantly reduce the accuracy of a radiometric identification scheme based on features extracted from the modulation domain by a commodity RF hardware platform.

Our Contributions We make the following key contributions in our work:

- Our work is, to the best of our knowledge, the first to consider and experimentally evaluate the ability of an active attacker to defeat a radiometric identification scheme by attempting to mimic the modulation error characteristics of another physical wireless device.
- Second, we consider radiometric identification in a more practical setting by evaluating its effectiveness using only a readily available software-defined radio platform. In doing so, we augment a previous modulation-based radiometric identification scheme with a feature that we found improves the accuracy of wireless transmitter classification performed by our SDR platform.

Outline The rest of this report is organized as follows. First, in Section 1.2, we will review the existing techniques for physical device fingerprinting. In Section 1.3, we describe the hardware and software components of the testbed we will use for evaluating the effectiveness of active attacks on modulation-based radiometric identification. A key aspect of our testbed is that it does not rely on expensive signal analyzers, opting instead for an inexpensive, consumer-level SDR platform. We also present a baseline evaluation that shows the accuracy of our radiometric identification implementation in the absence of adversarial interference.

Before describing the active attacks presented in this report, we will first review our threat model in Section 1.4. We then begin our attack analysis with a simple replay attack described in Section 1.5. Unlike traditional replay attacks, wherein simply the bytes in a wireless frame are captured and retransmitted, we implement a variation on the replay attack that attempts to sample and replay the actual waveform transmitted by a wireless transceiver. We show that the physical layer characteristics of a device are often sufficiently preserved in the replayed signal, allowing us to bypass radiometric identification a non-trivial percentage of the time.

Next, in Section 1.6 we present the results of an attack that attempts to characterize the modulation errors of a wireless transceiver using the constellation diagram observed for a transmitter by the attacker. The attacker then uses the observed constellation diagram to inject his own frames into the network that mimic the modulation error characteristics of the original transmitter.

We finally conclude in Section 1.7 with an overview of our results, a discussion of our results and the important limitations of our analysis, and some possibilities for future related research directions.

1.2 Background & Related Work

Previous work on wireless device identification and authentication has typically followed three general approaches: software-based fingerprinting, channel-based fingerprinting and hardware-based fingerprinting. We will review the techniques applicable to each one below, as well as their advantages and disadvantages.

1.2.1 Software-based Fingerprinting

The IEEE 802.11 protocol specification is both large and complex. Invariably, device manufacturers and driver developers will implement the protocol slightly differently while still being compliant with the specification. Using these variations in implementation, one can distinguish between not only devices made by different manufacturers but even between software revisions for the same device.

Franklin et al. [6] showed that the interval between probe requests sent by a wireless client varied between manufacturers. Ellch [12] took this line of work further and incorporated active testing by sending certain management frames, such as mangled association replies, to wireless clients and observing their responses.

An advantage of software-based fingerprinting is that it can be done completely passively, or actively for obtaining quicker results. Unfortunately, the granularity of identification using software-based approaches provides only for firmware-level identification at best. It does not allow one to distinguish between physical devices with the same software. Further, an adversary need only make some simple software modifications in order to mimic the behavior of another device.

1.2.2 Channel-based Fingerprinting

Trappe et al. [11, 16] sought to identify and characterize "forge resistant" relationships between pairs of wireless transceivers. The authors combined channel probing with periodic hypothesis testing to determine whether subsequent communication attempts are made by the same users as previous attempts. While this approach can determine if two communications are sent by the same user, it requires prior authentication in order to verify the identity of a communicant.

Faria and Cheriton [5] devised a scheme whereby wireless access points function as network sensors. The access points collaborate to use signal strength measurements in order to distinguish between clients located geographically apart.

The major disadvantage of this approach is that it essentially assumes wireless devices are immobile. If a device were to move, its observed *signalprint* thus changes. It is also unable to distinguish between two devices located in close proximity to each other.

1.2.3 Hardware-based Fingerprinting

The field of hardware-based fingerprinting is somewhat broad, even within the domain of wireless network security. We will identify and discuss the three most prevalent approaches.

Clock Skew Fingerprinting Modern computer hardware clocks are typically based on inexpensive crystal oscillators. Due to manufacturing variations, defects and even other environmental effects, the oscillator frequency can vary slightly between clocks of the same type. Kohno et al. [10] showed that these variations can be approximated based on TCP or ICMP timestamps and used to identify physical devices on a network.

Jana and Kasera [8] proposed using clock skew as a method for detecting rogue 802.11 wireless network access points. A wireless intrusion detection system (WIDS) can extract the timestamp values broadcasted by access points in beacon frames. After collecting a number of timestamp values, clients can estimate the clock skew of the access point. If the estimation varies from the expected clock skew of known legitimate access points, the WIDS node can alert an administrator to the possible presence of a rogue access point.

The current state of the art in clock skew fingerprinting, however, is not suitable for inclusion an authentication system. Clock skew fingerprinting relies on the device being fingerprinted returning the true value of its current clock. If an attacker controlling a rogue access point knows his clock skew relative to a legitimate access point, he can simply generate fake timestamps by adding or subtracting the appropriate offset thus appearing to have the same skew as the legitimate access point.

Physical Unclonable Functions Physical unclonable functions (PUFs) are a special type of hardware fingerprinting in that they are based on variations intentionally added to specially manufactured integrated circuits (ICs) contained within a device. As an example, an *arbiter PUF* is based on the delay of two signals through a series of MUXes that ends in an arbiter. The arbiter outputs a single bit whose value depends on relative delay between two signals racing through the series of MUXes [14]. The arbiter PUF will output a response unique to that PUF given a particular challenge as input. Someone validating the authenticity of the device containing the PUF can verify that the response is correct for the given challenge.

A fundamental disadvantage of PUFs, however, is the fact that they do require specially manufactured ICs in the network devices. Such hardware is not widely available and would be expensive or impossible to retrofit onto existing devices. Another disadvantage is that a large number of valid challenge-response pairs must be generated for each node prior to deployment and then distributed to all other nodes so that post-deployment authentication can be done.

Radiometric Identification Hall et al. [1] and Brik et al. [3] both proposed different techniques with the same goal of identifying wireless transceivers based on properties of the transmitted signals. Hall et al. focused on extracting the transient portion at the start of a modulated signal and then classifying the frame based on features extracted from the transient portion. Danev and Capkun also successfully used a transient-based approach in identifying nodes in a sensor network [4].

Brik et al. followed a slightly different approach with their PARADIS system, classifying frames instead based on features extracted from the modulation domain (e.g., average phase and magnitude errors). [3] suggested that one of the advantages of performing radiometric fingerprinting based on features extracted from the modulation domain was that its simplicity enables the scheme to be implemented by low-end devices, such as common wireless NICs or WLAN access points. Much like per-frame Radiotap or Prism headers are able to expose physical characteristics, such as channel frequency or RSSI values, to higher layer applications, a PARADIS-enabled NIC could include modulation error statistics like average phase and magnitude error. In evaluating the accuracy of PARADIS, though, the authors implemented their system using a high-end Agilent vector signal analyzer. Using such equipment, the authors showed their approach can yield around 99% accuracy in classifying frames based on five simple features extracted from a modulated 802.11b wireless frame.

In the remainder of this paper, we will further explore the use of features extracted from the modulation domain for radiometric fingerprinting. In particular, we will see that it is indeed possible for economy RF equipment to classify wireless frames based on a small set of features as the authors of PARADIS envisioned. We will also show that its simplicity in practical deployment environments enables an active attacker equipped with only modest hardware can reduce the accuracy of a modulation-based fingerprinting scheme by replaying or mimicking the physical characteristics of legitimate network devices.

1.3 Experimental Testbed

We had two main goals when constructing the testbed for our experiments. First, we wanted to replicate a potential real world deployment environment for a radiometric identification scheme that also enables us to easily experiment with active attacks against a modulation-based identification approach, such as that used by [3]. This meant ensuring that our software radio implementation must be able to interoperate with standards-compliant 802.11b equipment.

Second, we wanted our feature extraction and classifier implementation to model as closely as possible the approach described in earlier work [3] to allow for a fair comparison. It was not our goal to introduce a new method or approach for feature extraction and classification. Rather, we wanted to investigate the resilience of such a scheme against an active adversary under a potential deployment scenario.

1.3.1 Hardware Setup

We used two USRP2 hardware devices from Ettus Research, each of which was equipped with a RFX2400 transceiver daughterboard and a VERT2450 antenna. This hardware configuration allows us to transmit and receive within the 2.40 GHz to 2.48 GHz frequency band. The USRP2 contains dual 14-bit ADCs operating at 100MHz and dual 16-bit DACs operating at 400MHz. Each USRP2 transmitted complex baseband samples at 25MS/s to a host machine via a direct gigabit Ethernet link.

To replicate a potential deployment scenario, one USRP2 device acted as a radiometric identification sensor and another as an active attacker. The sensor's role was simply to sample the wireless channel, demodulate transmitted wireless frames, extract the features described next in Section 1.3.2 and record the results. The second USRP2 served as the active adversary in this context. Its behavior depended on the type of attack we evaluated (i.e., replay attack or frame injection attack). We elaborate more on the attack implementation in Sections 1.5 and 1.6.

In addition to the two USRP2 devices, we set up three additional identically-configured IBM ThinkPad T30 laptops to act as the legitimate or "honest" network devices. Each laptop contained an internal wireless adapter with an Intersil Prism 2.5 Wavelan chipset, and each was configured to act as an access point via the `hostapd` software operating on the 2.412GHz wireless channel. The reason for configuring each of the laptops as an access point was so they would broadcast beacon frames at regular intervals, which were then received and processed by one of the USRP2 devices. The test environment was located in an academic building with other wireless devices operating on the same channel, so real world non-adversarial interference was present during our experiments.

Our hardware platform used for RF signal analysis is certainly less sophisticated than those used in previous studies on RF fingerprinting [3, 7]. At the same time, we believe it is far more flexible, enabling us to quickly and easily evaluate and adapt the attacks presented in this paper. Given its modest cost and easily available hardware, we also believe it to be the most practical testbed and representative of hardware likely to be found in real world deployment environments unlike previous studies.

1.3.2 Software Implementation

We used the GNU Radio[1] signal processing library as the basis for the software portion of our implementation. GNU Radio is a free software project that has developed a framework for building software-defined radio systems. In traditional

[1] http://www.gnu.org/software/gnuradio/

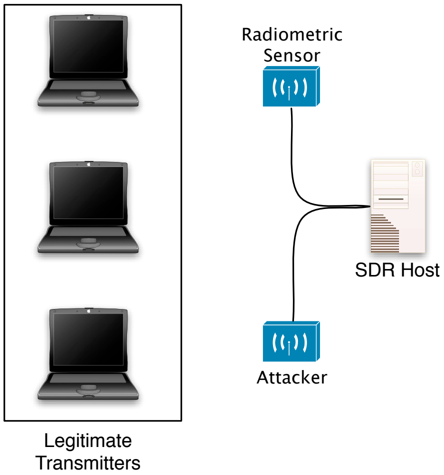


Fig. 1.1. Configuration of the testbed used in our evaluations. While the two software-defined radio devices were connected to the same host for signal processing, this was only for simplicity in conducting the experiments and no information was shared between the two devices.

radios, modulating, demodulating and processing waveforms is done almost exclusively in hardware. Software radio, on the other hand, processes a digitized signal using programmer-defined software code. Since the signal processing is implemented in software rather than purely hardware-based circuitry, a software radio framework is more modular and can easily be adapted for novel purposes.

Signal processing transforms are represented as a graph, with signal processing blocks of code represented as vertices and the edges between them represent the flow of data as the signal is processed. The signal is modeled as an infinite stream of data that flows from a signal processing block's input port, into the block where it is transformed and "pumped" to the block's output port. Any number of blocks can be combined to form a *flow graph* through which the sampled signal from the USRP2 hardware is processed. The flow graph is constructed using the Python programming language while the signal processing blocks themselves are implemented in C++.

For the 802.11b portion of our transceiver's software implementation we started with selected portions of existing code previously written for the original USRP device by BBN Technologies [2], but with several modifications to better take advantage of the more powerful USRP2 device and the most recent GNU Radio framework. Our code was also instrumented to enable the center frequency offset recovery and modulation feature extraction required for analyzing and classifying the processed wireless frames.

For feature extraction, we used the following modulation error characteristics, initially identified by the authors of [3]:

- *Frequency offset* — Offset from the ideal channel center frequency, f_c. The IEEE 802.11b specification tolerates center frequency offsets up to ±25ppm.
- *I/Q origin offset* — Distance between the ideal I/Q origin and the origin of a frame's observed I/Q constellation.
- *Average symbol magnitude error* — Difference between the ideal symbol magnitude and the received symbol magnitude, averaged over the entire frame. See Figure 1.2 for a depiction of symbol magnitude error.
- *Average symbol phase error* — Phase difference between the ideal symbol phase and the received symbol phase, averaged over the entire frame. See Figure 1.2 for a depiction of symbol phase error.

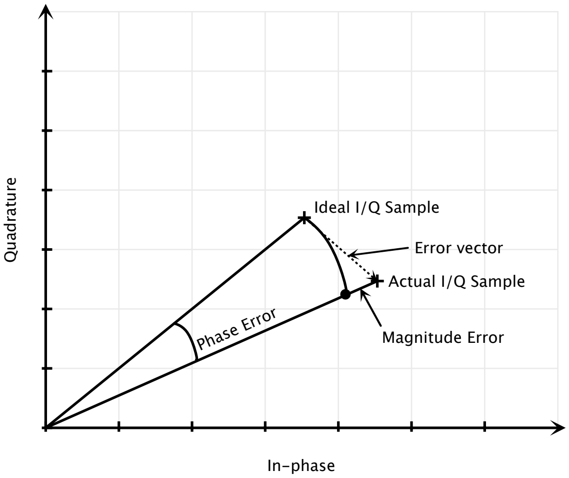


Fig. 1.2. Symbol magnitude error is the difference between the ideal magnitude of the I/Q signal and the received magnitude. Similarly, phase error describes the difference between the ideal phase of a symbol in a frame and that of the received symbol. Symbol magnitude and phase errors are averaged over all symbols in a single frame.

The authors of the Paradis also identified SYNC correlation as a useful fifth feature. SYNC correlation is defined as the normalized cross-correlation between the received wireless frame's synchronization (SYNC) preamble and the ideal modulation of the SYNC preamble. In implementing our testbed, we opted to replace the SYNC correlation feature with a simpler new feature we propose based on the deviation from the expected relative phase shift between two consecutive symbols, having found it easier to reliably extract given our available equipment as well as enhancing frame classification accuracy. We describe this feature in greater detail in Section 1.3.3.

Given the above set of five modulation features extracted from a transmitted frame, our implementation—like that in [3]—then uses a support vector machine (SVM) classifier to identify the most likely sender of the frame.

1.3.3 Baseline Evaluation

The first step in our experiments was to verify that our implementation was able to successfully distinguish between separate physical devices in our testbed without any adversarial interference. Further, we were interested in determining what, if any, reduction in accuracy we sustained by using an off-the-shelf software-defined radio platform instead of a special-purpose vector signal analyzer.

We first collected 500 beacon frames from each of the three identically configured laptops, as well as 500 beacon frames transmitted by the USRP2 that will be acting as the attacker in Sections 1.5 and 1.6. Each frame was processed as it was received and the sender's MAC address, frequency offset, average phase error, average magnitude error and I/Q offset for the frame. Any beacon frame with an invalid CRC value was discarded.

After collecting the per-frame statistics, we used a 5–fold cross-validation in evaluating the classification results over the 500 frames. k–fold cross-validation is a standard technique in machine learning that divides the input data in to k subsets, using a different subset each time as the training set and evaluating the model with the remaining $k-1$ subsets.

Using only the initial 4-tuple of features described above, we found the cross-validation resulted in a classification accuracy of only 66.7%. Given these results, it's clear that the baseline accuracy of the radiometric identification scheme diminished in our implementation from the 99% reported in [3], most likely as a result of our limited sampling accuracy. We expected a lower classification accuracy as a result of the lower resolution sampling capabilities and likely increased noise induced by our testbed's RF hardware, but still sought to improve our results.

In addition to the metrics described above we also added a new but related feature to the set, based on the relative phase shifts used to encode data bits. In phase shift keying (PSK), the phase of the signal itself compared to a known reference signal is used to convey information. For example, in binary phase shift keying (BPSK), a shift in phase of 180° from the reference signal may represent a bit value of 1, whereas no phase shift represents a bit value of 0.

In differential phase shift keying (DPSK), shifts in the phase between two consecutive symbols are used to convey information rather than the absolute phase itself relative to a reference signal. The degree of phase shift between two symbols determines the transmitted bit value or values. The advantage of differential DPSK over PSK in practical applications is that the transmitted signal can be demodulated without an additional carrier-recovery scheme (i.e., demodulation is *non-coherent*). Table 1.1 gives the expected phase shifts for 1.0 Mbps DBPSK and 2.0 Mbps DQPSK, respectively, as specified by Section 15.4.6.4 of the 802.11 standard.

Thus, in addition to the average symbol phase error described in Section 3.2, we also extracted the average symbol phase *shift* error computed over each frame. For example, if the expected phase shift to transmit a the dibit value `01` in DQPSK is $\pi/2$ and the actual phase shift relative to the previous symbol is

Bit Value	Phase shift
0	0
1	π

(a)

Dibit Value	Phase shift
00	0
01	$\pi/2$
11	π
10	$3\pi/2$

(b)

Table 1.1. 802.11b differential PSK encoding table for (a) DBPSK, and (b) DQPSK.

$\pi/2 + \epsilon$, the current symbol would have a phase shift error of ϵ. We computed the average ϵ over all symbols in a single frame. While this metric is quite clearly related to the absolute phase error of a single symbol, it is slightly different in that it characterizes the transition *between* all symbols in a frame rather than the error in the absolute phase of a single symbol. The intuition is that it is a simple way to roughly characterize instances of observed quadrature skew–that is, when the phase angle between the I and the Q vectors is not exactly 90 degrees.

After incorporating this feature into the classification process for the same 500 frames from the previous experiment, we saw the classification accuracy improve from 66.7% to 87.5%. Note that this value represents *average* classifier accuracy, and does not distinguish between false positives and false negatives. Rather, it simply represents the fraction of the total samples that were classified as belonging to the correct transmitter. In our evaluations of active attacks in the remainder of this paper, we will instead focus on false accept rates as that is the most clearly representative measure of success for an impersonation attack.

While our results are still significantly less than the accuracy in excess of 99% obtained by other work [3, 4, 9], we have showed that practical and reasonably reliable radiometric identification is indeed possible using only a commodity software-defined radio platform.

1.4 Threat Model

We will discuss the implementation and effectiveness of our attacks presently, but first we will more clearly define the network model, the adversary and the adversary's capabilities that we consider in the remainder of this paper.

1.4.1 Network Model

Wireless networks have a unique and particularly challenging threat model when compared to traditional wired networks. The broadcast nature inherent in a wireless channel means all communications sent and received can be overheard by an adversary within transmission range. The 802.11 body of standards have defined various supplementary security protocols that rely on supplementary keying information, such as WPA or 802.1x, but control and management frames are still sometimes sent unencrypted.

Due to the uncertain, unsupervised and potentially hostile deployment environment for wireless sensor networks, it is also reasonable and prudent to consider an active adversary with physical access to one or more network nodes. Under such a threat model, it becomes more untenable to rely on the secrecy of key material stored on the devices themselves to provide identification, such as in PKI-based approaches. It is in exactly such scenarios where radiometric identification provides a promising addition to existing authentication techniques, since it can be done without relying on secret keying material that may become compromised by an active adversary.

In line with previous work on radiometric identification [3, 1], we assume the presence of one or more *radiometric sensors* whose responsibility it is to detect rogue devices in the network. As wireless frames are transmitted, the radiometric sensors extract one or more features from the transmitted signal in order to compute the *radiometric signature* for the transmitting device. If the computed signature does not match the known signature for a legitimate wireless device in the network, the radiometric sensor will alert an administrator or other nodes in the network to the presence of a rogue device.

1.4.2 Adversarial Capabilities

We assume the adversary possesses an RF transceiver capable of operating within the same frequency range as the legitimate devices in the wireless network. The adversary is thus able to not only receive any wireless frames sent by other devices within transmission range, but he is also able to inject his own traffic into the network. The adversary may simply retransmit (or *replay*) frames previously sent by legitimate transmitters, or he may construct and transmit arbitrary wireless frames as well. In contrast with earlier work, we introduce and consider an active adversary who purposely tries to modify his radiometric signature in order to evade detection by the network's radiometric sensors.

Of course, it is also feasible for an active adversary to simply "jam" the network by constantly broadcasting wireless signals and preventing any legitimate traffic from being transmitted and thus preventing any node from authenticating another. While jamming resistance is an active area of research in itself, we do not consider such a scenario further in this work.

1.5 Replay Attacks

Having described our adversarial model and determined the baseline accuracy of our radiometric identification implementation, we now move on to evaluating the potential impact of active attacks against our testbed. We begin our analysis by implementing a simple replay attack.

1.5.1 Attack Description

The replay attack we implemented has three main steps, each described below.

1. *Capture and extract the frame to be replayed*
   First, we use a USRP2 to capture beacon frames transmitted by each of the three test laptops. Rather than processing the received signal, we simply save all of the captured baseband I/Q samples to disk at 25MS/s. The file of captured samples was then filtered in order to extract only the sets of I/Q values corresponding to a transmitted frame from the rest that merely contain channel noise. To do this, we used a simple exponentially weighted moving average (EWMA)-based envelope estimation mechanism to quickly estimate the start and end of a transmission within the set of all captured samples. The signal envelope $E(t)$ for the complex baseband signal was estimated as

   $$E(t) = E(t-1) \cdot (1-\alpha) + \frac{\alpha}{2} \cdot (|I(t)| + |Q(t)|), \tag{1.1}$$

   where α acts as the "smoothing" factor, and $I(t)$ and $Q(t)$ are the in-phase and quadrature components of the complex sample. When $E(t)$ exceeds some threshold τ, then the subsequent samples are processed by our PHY layer implementation until $E(t)$ falls below τ again. If the processed frame is valid and has a correct CRC value, the samples corresponding to that frame are saved to a separate file. The parsed sets of frame samples were then grouped according to the transmitter's MAC address parsed from the demodulated frame.
2. *Determine the corrected center frequency offset*
   One of the features used to classify captured frames is channel center frequency offset. It is a measure of the frequency offset by which the receiver had to be adjusted in order to obtain carrier lock. The authors of [3] found it to be the feature that contributed the most towards an accurate classification result; however, we also found carrier frequency offset to be an easy and straightforward feature for an attacker to impersonate.
   We first identified the frequency offset of the attacker's transmitter by transmitting a constant carrier at the ideal channel center frequency f_c and measuring the frequency of the actual transmitted carrier $\tilde{f}_c$ from a reference receiver. The center frequency offset of the attacker f_o was then simply

   $$f_o = \tilde{f}_c - f_c.$$

   Let $\tilde{f}_o$ be the observed frequency offset of the target device whose radiometric signature we are attempting to impersonate. When injecting wireless frames into the network, the center frequency of the attacker's transmitter is adjusted to $\hat{f}_c$, where

   $$\hat{f}_c = f_c - f_o + \tilde{f}_o.$$

   The corrected center frequency $\hat{f}_c$ compensates for both the attacker's own frequency offset, as well as that of the impersonation target's device.
3. *Replay the captured frame samples*

> The attacker's transmitter is first tuned to the corrected carrier frequency $\hat{f}_c$. Next, the baseband I/Q samples corresponding to the frame we want to replay are read from the capture file and then sent from the host to the attacker's USRP2, where they are upconverted to RF and transmitted as a carrier signal modulated according to the previously captured and extracted I/Q samples.

Thus, rather than simply replaying bytes transmitted by a legitimate device, we are actually replaying I/Q samples captured during the replayed frame's original transmission. Due to errors introduced by both attacker's sample capture, extraction and retransmission process, as well as external factors such as interference and multipath effects, the replayed waveform will most certainly not be identical to the original. Our present evaluation shows, however, that its error characteristics were sufficiently preserved, allowing a majority of the replayed frames to be misclassified by our USRP2-based implementation as having been transmitted by the original device.

1.5.2 Evaluation

Using one USRP2 we captured and extracted the complex samples corresponding the 500 beacon frames for each of the three laptops acting as legitimate transmitters using the process described above. We then retransmitted the captured samples using the attacker USRP2 and captured them again with the second USRP2. The modulation error statistics for both the set of original frames and the frames retransmitted by the attacker were captured and saved. Again, any frames that did not meet the 802.11b specifications were discarded and not included in the analysis.

We then trained the SVM classifier with 50 beacon frames sent by each of the three legitimate device, and an additional 50 beacon frames transmitted by the attacker. Since the goal is to determine if replaying the complex samples captured from another transmitter lets the attacker sufficiently alter his radiometric fingerprint, the beacon frames used to train the classifier for the attacker were modulated and transmitted according to the 802.11b standard. No attempt was made to alter the attacking USRP2's radiometric signature, since doing so would have distorted the classifier's training phase.

We then attempted to classify 500 beacon frames from each of the three transmitters that had been replayed by the attacker. Classification testing was again done using 500 frames from each legitimate transmitter that had been replayed by the attacker, and captured by the other device performing the radiometric identification and classification. The result of this experiment was a false accept rate of approximately 75%. We see that the replay attack was not perfectly reliable for the attacker, unlike many forms of replay attacks which simply replay frame bytes; however, it did have a significant measure of success in defeating our implementation of a modulation-based radiometric identification scheme.

1.5.3 Discussion

While the replay attack above is simple and straightforward, the implications of showing its reasonable success against even radiometric identification techniques is important. Consider, for example, wormhole attacks [13]. In a wormhole attack, the adversary captures transmitted data at one location, sends it over a separate attacker-controlled link (the "wormhole"), and then replays it at the other end of the link. Devices at one end of the wormhole thus believe they are within transmission range of devices transmitting at the other end.

Radiometric identification could be useful in such instances, in order to detect the adversary-controlled transmitter; however, we have shown that, rather than simply relaying packet data, theoretically the attacker can sample the transmitted waveform of another device in the network, relay those samples across the wormhole, and then replay those samples at the other side. If the adversary is equipped with an accurate enough receiver and corresponding transmitter, devices performing radiometric identification at one end of the wormhole are likely unable to distinguish between the adversary's transmitter and a legitimate transmitter located at the other end of the wormhole.

1.6 Frame Injection Attacks

The replay attack presented above allows an adversary to replay a previously captured wireless frame sent by a legitimate transmitter, while mimicking the radiometric signature of the frame's original sender; however, it does not allow the adversary to generate and insert his own traffic into the network. In this section, we will explore a variation on the attack that allows the adversary to generate arbitrary frame content and inject it into the network while still hiding his own radiometric signature.

1.6.1 Attack Description

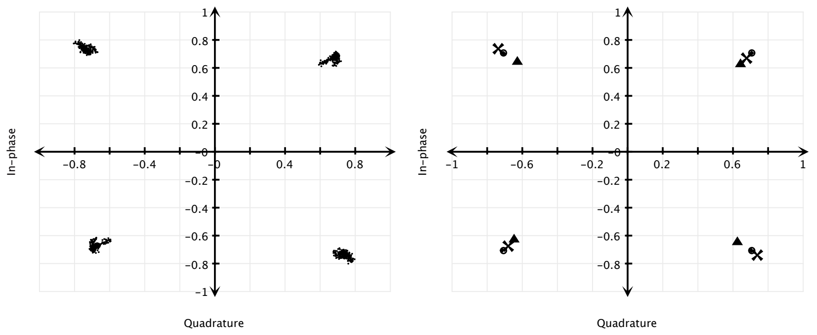
Our attack is based on extracting two main properties from a device's radiometric signature, and using those to adjust the behavior of the attacker's transmitter. We discuss how to define each of these properties below.

1. *Determine the corrected center frequency offset*
   As in the replay attack description in Section 1.5, the attacker first learns the average center frequency offset of the target device by observing a number of frames transmitted by the target. The attacker also learns his own transmitter's center frequency offset with the cooperation of a reference receiver he controls.
2. *Derive the observed PSK constellation*
   Recall the five characteristics of a modulated signal that we use for feature extraction: center frequency offset, average symbol magnitude error, average symbol phase error, average inter-symbol phase shift error and I/Q origin

offset. With the exception of the center frequency offset, all of these features characterize the degree to which the actual observed symbol constellation differs from the ideal constellation given in the 802.11 specification. Thus, the premise of this step in the attack is to derive an approximate PSK constellation that the attacker will then use to modulate wireless frames injected into the network.

To derive the constellation used to impersonate another device, the attacker starts by collecting 50 frames transmitted by the target device. The I/Q points from each frame are extracted, forming clusters near the ideal I/Q constellation points as shown in Figure 1.3(a). We also note that the sampled I/Q points in Figure 1.3(a) are quite noisy, which can help explain the decreased accuracy of our implementation over an implementation using a high-end signal analyzer [3].

The center for each symbol cluster is then computed, resulting in an approximation of the received constellation. An example of the constellations derived for each of the three legitimate transmitters in our testbed is given in Figure 1.3(b).



(a) (b)

Fig. 1.3. (a) An example of a subset of I/Q points sampled from a single frame transmitted by a single laptop that appears to be exhibiting slight quadrature skew, and (b) Derived QPSK constellation diagrams for the three different transmitters used in our testbed, compared to the ideal constellation.

3. *Transmit injected frame with derived PSK constellation*
   As in the replay attack, the attacker's transmitter is tuned to the corrected center frequency offset, $\hat{f}_c$, computed in Step 1. The attacker then simply transmits the frame he wants to inject into the network, modulated according to the PSK constellation derived in the previous step.

1.6.2 Evaluation

We evaluated the success of our frame injection attack much like we did for the replay attack above. We started by training the SVM classifier with 50 beacon frames from each of the three transmitters and the attacker's USRP2. PSK constellations for each of the three legitimate transmitters were derived from the 50 frames from each transmitter used to train the classifier. The results of this step are shown in Figure 1.3(b).

The attacker then sent 500 forged beacon frames for each legitimate transmitter, which were modulated according to the derived constellation for the target device. The forged frames were then received by the radiometric sensor and stored for later processing. If a frame was received with a bad CRC value, it was discarded and an additional frame would be sent by the attacker. In our evaluation, we found that around 10% of the forged frames received by the radiometric sensor had to be discarded because of an invalid checksum.

The 1500 successfully received forged frames were then evaluated using the SVM classifier. We found that the classifier resulted in a combined false acceptance rate of 55%. In other words, only around half of the packets sent by the attacker were able to successfully replicate the features of the target device accurately enough to evade radiometric identification. The remaining frames were correctly identified as not having been sent by the attacker's target device.

1.6.3 Discussion

We first note that the 55% success rate in our frame injection attack is markedly less than the 75% success rate in our frame replay attack; however, this reduction is not surprising. We do, however, see several possibilities for an ambitious attacker to improve the success rate of a frame injection attack. In particular, we have not yet explored the possibility of characterizing more aspects of the attacker's own transmitter imperfections (e.g., I/Q offset and phase errors) and compensating for them when injecting frames into the network under the guise of another device's radiometric signature. Doing so would likely improve the attacker's results under both the replay attack and the frame injection attack, but we will leave this aspect to future work.

We may also improve results by implementing an approach similar to a hill-climbing attack against biometric authentication systems [15]. In a hill-climbing attack on a fingerprint recognition system, the attacker slightly alters the minutiae on a synthetically generated template and submits it to a matcher. Based on the result, he modifies the template again, attempting to improve the match score each time until he exceeds the threshold required for acceptance.

Similarly, an active attacker in our scenario may slightly modify the constellation used to modulate the injected frame until its features more closely match those of the target device. As we have shown above, an attacker using a software-defined radio platform can enable the attacker to quickly and easily redefine such properties of the modulated signal. Again, we leave this as a potential direction for future work.

For now, we simply have sought to show that it is at least *possible* to impersonate another device's radiometric signature when feature extraction and classification is done in the modulation domain. In doing so, we hope to draw attention to the fact that a modulation-based radiometric identification scheme, while simple and efficient, may not offer the best security against active attacks.

1.7 Conclusions

Existing work in the area of RF fingerprinting and identification has implicitly made the assumption that the minute physical characteristics of physical devices that result from variations and defects in manufacturing process are unforgeable. Such work has largely focused on maximizing their ability to correctly distinguish between unmodified wireless network devices that make no attempt to alter or disguise their own RF signature.

In this report, we have shown how a software-defined radio platform can likely be leveraged in order to implement a practical radiometric identification scheme. Even though we are using more basic hardware than previous studies, we are still able to achieve over 87% accuracy in 802.11b transceiver identification. Our use of a software-defined radio platform also allowed us to bring to attention the potential threat active attackers can pose to a modulation-based radiometric identification scheme by mimicking such physical layer characteristics of other devices. We showed that a basic replay attack is able to achieve a success rate of over 75% against our implementation by retransmitting a sampled complex signal rather than simply replaying packet bytes. Further, we showed that such an attacker can not only replay previously sent frames, but also inject his own arbitrary traffic into the network and evade detection slightly more than 50% of the time.

While our preliminary work represents a step in the right direction towards properly considering the security of RF fingerprinting under a more realistic threat model, it is critical to keep our initial results in perspective. We reiterate that the evaluation of our attacks was conducted with a less sophisticated testbed than in previous work. This was done first to evaluate earlier speculation in [3] that an approach based on the modulation domain was more practical and efficient, which we indeed found to be true. But the SDR-based implementation also allowed us to evaluate what the security consequences are in moving to a more practical hardware platform for an identification scheme, which has often been overlooked by recent work in the field.

It may be argued that a high-end vector signal analyzer capable of a much greater sampling accuracy would be able to successfully differentiate between legitimate frames and those transmitted by the attacker in our USRP2-based testbed. Indeed, we believe this is very likely to be the case; however, we could also make the counter-argument that the simple impersonation techniques examined in this paper could similarly be replicated using a much higher-quality signal generator and perhaps still succeed against a special-purpose signal analyzer. Consequently, the contention between a modulation-based radiometric identification scheme and an active attacker may simply be an “arms race” with respect to who can afford the highest quality equipment. Investigating this aspect further and more formally is the subject of our future work.

It is also important to note that we have not evaluated the effectiveness of the attacks presented above against RF fingerprinting techniques that are based on extracting features from the transient portion of a signal, rather than on features extracted from the modulation domain. We intend to pursue this direction in future work, but we currently believe that transient-based schemes are likely to be immune to the simplistic active attacks presented in this paper. At the same time, a transient-based identification scheme is less efficient and more difficult to implement than a modulation-based scheme. Thus, we expect there to be a tradeoff between the strength and accuracy of a radiometric identification scheme, and its ability to be implemented with basic hardware. In future work, it may be beneficial to more formally quantify such tradeoffs. Still, our preliminary work here has made a necessary first step towards understanding the limits of modulation-based radiometric identification under practical deployment scenarios and adversarial models.

Acknowledgments

This work was supported by the National Science Foundation under award number CNS-0831366.

References

1. M. Barbeau, J. Hall, and E. Kranakis. Detecting impersonation attacks in future wireless and mobile networks. In *Proceedings of Secure Mobile Ad-hoc Networks and Sensors (MADNES 2005)*, pages 80–95, 2005.
2. BBN Technologies. GNU Radio-based 802.11b implementation. `https://www.cgran.org/wiki/BBN80211`, Accessed February 2009.
3. V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *MobiCom ’08: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, pages 116–127, New York, NY, USA, 2008. ACM.
4. B. Danev and S. Capkun. Transient-based identification of wireless sensor nodes. In *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) 2009*, 2009.
5. D. B. Faria and D. R. Cheriton. Detecting identity-based attacks in wireless networks using signalprints. In *WiSe ’06: Proceedings of the 5th ACM workshop on Wireless security*, pages 43–52, New York, NY, USA, 2006. ACM.
6. J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *USENIX-SS’06: Proceedings of the 15th USENIX Security Symposium*, Berkeley, CA, USA, 2006. USENIX Association.

7. J. Hall. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. In *In Proceedings of the 3rd IASTED International Conference on Communications, Internet and Information Technology (CIIT)*, pages 201–206. Kranakis, 2004.
8. S. Jana and S. K. Kasera. On fast and accurate detection of unauthorized wireless access points using clock skews. In *MobiCom '08: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, pages 104–115, New York, NY, USA, 2008. ACM.
9. I. O. Kennedy, P. Scanlon, F. J. Mullany, M. M. Buddhikot, and K. E. Nolan. Radio transmitter fingerprinting: A steady state frequency domain approach. In *2008 IEEE Vehicular Technology Conference*, pages 1–5, September 2008.
10. T. Kohno, A. Broido, and K. C. Claffy. Remote physical device fingerprinting. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 211–225. IEEE Computer Society, 2005.
11. Z. Li, W. Xu, R. Miller, and W. Trappe. Securing wireless systems via lower layer enforcements. In *WiSe '06: Proceedings of the 5th ACM Workshop on Wireless Security*, pages 33–42. ACM Press, 2006.
12. J. P.Ellch. Fingerprinting 802.11 devices. Master's thesis, U.S. Naval Postgraduate School, September 2006.
13. R. Poovendran and L. Lazos. A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. *Wireless Networks*, 13(1):27–59, January 2007.
14. G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *DAC '07: Proceedings of the 44th annual conference on Design automation*, pages 9–14, New York, NY, USA, 2007. ACM.
15. U. Uludag and A. Jain. Attacks on biometric systems: A case study in fingerprints. In *Proceedings of SPIE-EI 2004, Security, Steganography and Watermarking of Multimedia Contents*, volume 5306, pages 622–633, 2004.
16. L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe. Fingerprints in the ether: Using the physical layer for wireless authentication. *ICC '07: 2007 IEEE International Conference on Communications*, pages 4646–4651, June 2007.