This paper was presented as part of the main technical program at IEEE INFOCOM 2011

Device Fingerprinting to Enhance Wireless Security using Nonparametric Bayesian Method

Nam Tuan Nguyen, Guanbo Zheng, Zhu Han and Rong Zheng‡,
ECE Department, ‡CS Department, University of Houston, Houston, TX 77004

***Abstract*—Each wireless device has its unique fingerprint, which can be utilized for device identification and intrusion detection. Most existing literature employs supervised learning techniques and assumes the number of devices is known. In this paper, based on device-dependent channel-invariant radio-metrics, we propose a non-parametric Bayesian method to detect the number of devices as well as classify multiple devices in a unsupervised passive manner. Specifically, the infinite Gaussian mixture model is used and a modified collapsed Gibbs sampling method is proposed. Sybil attacks and Masquerade attacks are investigated. We have proven the effectiveness of the proposed method by both simulation data and experimental measurements obtained by USRP2 and Zigbee devices.**

I. Introduction

Due to the broadcast nature of wireless medium and programmability of wireless devices, identity of legitimate wireless devices can be easily spoofed. For instance, an *ioctl* system call in the Linux kernel can modify the MAC address of a network interface card. Modifying or replacing the EPROM in a phone would allow the configuration of any ESN (Electronic Serial Number) and MIN (Mobile Identification Number) via software for cellular devices. Once the device identity is compromised, multiple attackers can masquerade as a single legitimate user, or a single attacker can assume multiple legitimate identities. In both cases, many forms of attacks can be launched including Sybil attacks [1], masquerade attacks [2], resource depletion [2], traffic injection [3], denial of service [4], etc. These attacks may result in spurious outage of communication and leakage of critical information. Therefore, it is crucial to detect the presence of identity spoofing, and/or determine the number of attackers involved.

Existing solutions to the detection of identity spoofing roughly fall into two categories, namely, active detection and passive detection. In active detection, additional messages are exchanged among many network entities. For instance, cryptographic based schemes can be used to either authenticate users or elicit user specific responses [5]. To handle the case that the entire node is compromised or captured (and thus cryptographic keys are exposed), location dependent information can be utilized to assist the authentication process. The basic premise is that a physical device cannot appear in more than one locations at the same time. In [6], a simple active method is proposed for discovering facts about the chipset, the firmware or the driver of an 802.11 wireless device by observing its responses (or lack thereof) to a series of crafted non-standard or malformed 802.11 frames. Active detection methods require extra message exchanges and thus can accelerate energy consumption or reduce throughput of legitimate devices. Furthermore, responses that are firmware, driver, OS dependent can be spoofed as well. In contrast, passive detection methods extract device or location specific features from message transmissions. Device specific features typically include, clock skew (observed from message time stamps), sequence number anomalies (in MAC frames), timing (of probe frames for channel scanning), and various RF parameters (transient phases at the onset of transmissions, frequency offsets, phase offsets, I/Q offsets etc.). Location dependent features are typically radio signal strength (RSS) vectors measured at trusted devices. Location based features cannot identify devices but when combined with device IDs, it is useful in distinguishing devices with the same ID. The disadvantage of location based features is that RSS measurements are time varying and have low spatial resolutions. Devices a few meters apart in an indoor environment may have similar RSS features.

In this paper, we propose to use device dependent radio-metrics as fingerprints to detect identity spoofing. A radio-metric is a component of radio signal such as amplitude, frequency, phase or any feature derived from those components. Each device creates a unique set of radio-metrics in its emitted signal due to hardware variability during the production of the antennas, power amplifiers, ADC, DAC, etc. As a result, radio-metrics cannot be altered post-production and thus provide a reliable means for identification. Choosing which radio-metrics to form the device fingerprints is technology dependent. Different modulation techniques have different sets of radio-metrics. For example, in Quadrature Phase Shift Keying (QPSK), the characteristics of all four symbols can be utilized to form a constellation. From the constellation, some idiosyncratic features can be extracted and used as the device fingerprints [10].

Our approach differs from existing work in the following aspects. First, the features selected are channel invariant, and are thus insensitive to transmitter/receiver antenna gain, distance and moderate mobility. Second, it is a passive detection method that does not require out-band message exchanges. Third, arguably most importantly, our method is based on an unsupervised clustering approach. Unlike [6], [7], [8], there is no need to register legitimate devices and obtain training sequences to set up a database of feature space of legitimate devices. Furthermore, it does not assume *a priori* knowledge regarding the number of devices present in the networks. We model the feature space of a single device as a multi-variable Gaussian distribution with unknown parameters, and that of multiple devices (of the same or different device IDs) as an infinite Gaussian mixture (though only a finite subset are observed). We develop a non-parametric Bayesian approach

978-1-4244-9921-2/11/$26.00 ©2011 IEEE

to unsupervised clustering with an unbounded number of mixtures. Specifically, we define a prior over the likelihood of devices using the Dirichlet distribution [9] . Based on the properties of the Dirichlet distribution, we derive a Gibbs sampling algorithm that can be used to sample from the posterior distribution, and determine the number of clusters. Identity spoofing is determined by comparing the cluster labels with the device IDs. When there are multiple devices sharing the same device ID, the masquerade attack is identified. On the other hand, if several device IDs map to a single cluster, the Sybil attack is detected.

The effectiveness of the proposed method is validated using both simulated as well as measurement data. To obtain measurement data, we use a USRP2[11] receiver to collect the radio-metrics from 4 Zigbee transmitters at different time of the day and at different distances from the receiver. We find that the proposed scheme has high probabilities of detecting attacks and identifying malicious nodes.

It is worth mentioning that the proposed method is not limited to the set of radio-metrics evaluated in this paper. Incorporating additional device dependent features in the clustering method is expected to improve the detection accuracy, and will be explored in our future work.

The rest of the paper is organized as follows. In Section II, we review the related work. In Section III, the system model for wireless device fingerprinting is given. In Section IV, the nonparametric Bayesian method is investigated. In Section V, the algorithm for intrusion detection is constructed. Simulation results are shown in Section VI and the conclusion is drawn in Section VII.

II. Related Work

In this section, we will review the most relevant work in literature. Given the advantages of hardware-based fingerprinting compared to software-based fingerprinting, and passive detection methods over active detection methods, we focus on passive detection methods using radio frequency fingerprinting and location fingerprinting. Depending on whether training phases are involved, we can further classify these methods into supervised and unsupervised classification. In supervised methods, an authentication station maintains a "white list" of legitimate devices while a unsupervised method does not require legitimate devices to be registered first.

A. Supervised Classification Method

In [8], the authors proposed a Passive RAdio-metric Device Identification System (PARADIS) with an accuracy of detection over 99%. The results shows convincingly that radio-metric can be used effectively to differentiate wireless devices. Nonetheless, the method requires a training phase to collect and extract fingerprints of legitimate devices.

Hall et al. [7] proposed another method to identify devices using the transient signals at the start of transmissions. Although there is a very high correlation between the transient signal and the device, the transient signal is difficult to capture since it lasts on the order of only several hundreds of nanoseconds.

B. Unsupervised Classification Method

Compared with supervised methods, unsupervised method has the advantage of differentiating devices without the knowledge of device fingerprints ahead of time. Most of the previous works in this area are location-based methods. In [2], the received signal strength (RSS) vector of devices are measured by surrounding Access Points (APs). The RSS vectors have some functional relationship with the environment and the locations of devices. The method suffers from two problems. First, stationary device locations are implicitly assumed. Second, due to the time varying nature of wireless channels, devices at close proximity may have similar fingerprints, or a single device may have different fingerprints over time. Yang et al. [12] extend the idea of location-based fingerprinting by recording the signal strength over time and collect multiple RSS value vectors. They then proposed an algorithm to classify the multiple RSS vectors in a multidimensional space, which allows determination of the number of devices. It suffers from the weakness of location-based features as discussed above. Furthermore, in counting the number of devices, the classification method entirely relies on distance between the RSS vectors and fails to take into account valuable prior information regarding fingerprints.

Our method is based on more reliable features, the radio-metrics, and employs a nonparametric Bayesian model, an advanced statistical method in classifying the features. We do not assume prior knowledge regarding the number of devices, and thus can cope with scenarios where devices move in and out of the monitored space at run time.

III. System Model for Device Fingerprint

In this paper, we limit our attention to wireless technologies that employ variants of Quadrature Phase Shift Keying (QPSK) modulation schemes at the PHY layer. Examples are the IEEE 802.15.4 (Zigbee), IEEE 802.11, Bluetooth, etc. In QPSK, four symbols with different phases are transmitted and each symbol is encoded with two bits. The transmit symbols can be represented as follow:

$$s_i(t) = \sqrt{\frac{2E_s}{T}} \cos\left[2pif_ct + (2n-1)\frac{pi}{4}\right], n = 1, 2, 3, 4, \tag{1}$$

where E_s is the transmission power, T is symbol period, f_c is the carrier frequency and n is the index for the four possible constellations. By changing n, we can vary the phases of the signal, creating four phases $pi/4$, $3pi/4$, $5pi/4$, and $7pi/4$. In the OQPSK scheme used by Zigbee radios, by offsetting the timing of the odd and even bits by one bit-period, or half a symbol-period, the in-phase and quadrature components never change at the same time.

To distinguish wireless devices, we need to find the fingerprints, which is a group of unique features of devices. These features are extracted from the transmitted signals. The main features of any signal include the frequency, phase, and amplitude. It is worth mentioning that the transmitter fingerprints are different from the receiver's radio-metric parameters such as the received power. The transmitter fingerprints are uniquely determined by the transmitter's characteristics, and are not

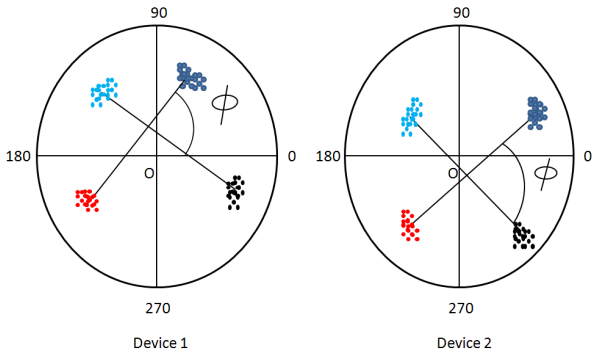


Fig. 1. Phase shift difference in the constellation of two Tx devices

affected by the channel and receiver design. For example, the received power is affected by both channel and receiver antenna gain, and thus is not a suitable feature.

The following two transmitter features extracted from (1) are collected to form our data set.

- **Frequency difference** δf_c: The difference between the carrier frequency of the ideal signal and the one of the transmitted signal. For different wireless transmitter devices, the carrier frequency f_c is likely to be different. δf_c can be measured from the frequency offset of the Phase Lock Loop at a coherence receiver.
- **Phase shift difference** ϕ: In the ideal case, the phase shift from one constellation to a neighbor one is 90^o. However, the transmitter amplifiers for I-phase and Q-phase might be different. Consequently, the degree shift can have some variances. Figure 1 shows an illustrative example of device signal constellations. The constellation may deviate from its original position due to hardware variability, and different devices have different constellations. The angle ϕ is employed as one feature in our classification method. Due to the possible I/Q ambiguity in QPSK, we consider the smaller of the two possible angles (ϕ or $pi - \phi$).

The proposed schemes in this paper can be easily extended to incorporate other features. The two above features corresponds a feature point in the 2-dimensional feature space. The feature points recorded over time from the same device should concentrate in a region in the feature space. Those from different devices shall form different clusters. The information on the number of clusters, or ideally the number of unique devices is valuable in detecting identity spoofing. For instance, if multiple cluster labels map to a single device ID, then we can detect the existence of masquerade attacks. On the other hand, Sybil attacks can be identified if a single class label maps to multiple device IDs.

Note that a *cluster* is modeled as a distribution with a certain unique parameter set. Or equivalently, a parameter set represents a unique *class*. Thus, in this paper, the two terms are used interchangeably depending on different contexts.

IV. Nonparametric Bayesian Model

Given the feature points from the feature space, the main questions are i) *how many devices have generated the data* and *which device each feature point belongs to*. To answer these questions, we first propose a way, in which the feature points are generated or in other words, a way to model the feature points. Depending on our prior knowledge about the data, two classes of models can be used. The Finite Mixture Model (FMM) is used in the case where the number of clusters is known a priori, while the Infinite Mixture Model (IMM) is used in the situation where the number of clusters is unknown or may vary over time. Both models are also called generative models, from which data are generated. After establishing the models, we will use them to develop algorithms to cluster the feature space in the next section.

Note that although the IMM is adopted in our clustering approach, the IMM is built upon the FMM, and thus the understanding of the later is essential.

A. Dirichlet Distribution

The Dirichlet distribution is the multivariate generalization of the beta distribution, and the conjugate prior of the categorical distribution and multinomial distribution in Bayesian statistics [9]. In other words, its probability density function represents the belief that the probabilities of K rival events $(\pi_1, \ldots, \pi_K)$, given that event π_i has been observed $\alpha_i - 1$ times, $i = 1, \ldots, K$.

Definition 1: The Dirichlet distribution of order $K \geq 2$ with parameters $\alpha_1, \ldots, \alpha_K > 0$ has a probability density function with respect to the Lebesgue measure on the Euclidean space $\mathbb{R}^{K-1}$ given by

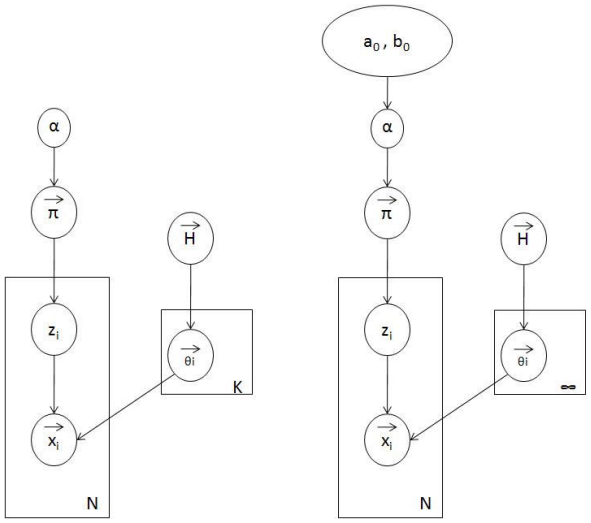
$$\text{Dir}(\alpha_1, \ldots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} \pi_i^{\alpha_1 - 1} \tag{2}$$

for all $\pi_1, \ldots, \pi_K > 0$ satisfying $\pi_1 + \cdots + \pi_K = 1$. The density is zero outside this open $(K-1)$-dimensional simplex. The normalizing constant is written in terms of the multinomial beta function, which can be expressed in terms of the gamma function:

$$B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)}. \tag{3}$$

B. Finite Mixture Model

We first start assuming that the number of classes is known. When the distribution is Gaussian, the Finite Mixture Model becomes the Finite Gaussian Mixture Model (FGMM). Define a matrix $\vec{X} = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_N]$, the data set of N feature points, in which each $\vec{x}_i$ is a vector of D dimensions. In our application, we consider frequency difference δf_c and phase shift difference ϕ. Thus, $D = 2$. In Figure 2(a), $\vec{\pi}$ is the mixing weight vector, $[\pi_1, \pi_2, ..., \pi_K]$, and represents the probability of assigning one feature point to one of the classes. K is the number of classes. Each class has its own distribution and $\vec{\theta}_k$ is the parameters for that distribution. Because we assume the distribution of the observations is Gaussian, each $\vec{\theta}_k$ consists of $\vec{\mu}_k$ and Σ_k. $\vec{H}$ are the hyper-parameters, which are parameters of the distribution of the parameters $\vec{\theta}_k$'s. The hyperparameters represent our knowledge of the observations. The rectangular shape denotes the repeated sub-structures. Each indicator, z_i, is associated with a feature point $\vec{x}_i$ indicating which class it



(a) Finite (b) Infinite

Fig. 2. Gaussian Mixture Model

belongs to. Specifically, $z_i = k$ indicates that $\vec{x}_i$ belongs to class k with probability:

$$p(z_i = k) = \pi_k. \tag{4}$$

According to [13], FGMM can be defined as bellow.

Definition 2: Finite Gaussian Mixture Model is defines as

$$\begin{aligned} \pi|\alpha &\sim Dir(\vec{M}) \\ z_i|\pi &\sim \text{Multinomial}(\cdot|\vec{\pi}) \\ \vec{\theta}_k &\sim \vec{H} \\ \vec{x}_i|\vec{\theta}_k &\sim \text{Gaussian}(\cdot|\vec{\theta}_k) \end{aligned} \tag{5}$$

where $\vec{M} = [\alpha/K, \alpha/K, ..., \alpha/K]$. K is finite.
The main challenge in applying FGMM is model selection, i.e., how to determine the number of classes, K. In our problem, if we know exactly the number of wireless devices then the number of clusters in the feature space is known. Unfortunately, in most situations, we have no control of the number of active (legitimate or illegitimate) devices. Therefore, the FGMM cannot model the feature space, and a more advanced statistical method is needed.

C. Infinite Gaussian Mixture Model

When the number of classes is unknown or can vary as more data is observed, we can model the feature space using Non-Parametric Bayesian (NPB) Models. A particular example of the NPB model is the IGMM. The IGMM is an extension of FGMM by letting $K \to \infty$. It is assumed that the number of classes or devices is infinite but only a finite number of devices are active at a certain time. As the result, only a finite number of classes are observed.

Figure 2(b) gives a graphical representation of the IGMM. The IGMM is identical to FGMM except that the number of classes is infinity. This difference leads to variations between the two models as seen in the definition bellow. With similar notations as in FGMM, the IGMM is defined as:

Definition 3: Infinite Gaussian Mixture Model.

$$\begin{aligned} \vec{\pi}|\alpha &\sim \text{Stick}(\alpha) && (6)\\ z_i|\vec{\pi} &\sim \text{Multinomial}(\cdot|\vec{\pi}) && (7)\\ \vec{\theta}_k &\sim \vec{H} && (8)\\ \vec{x}_i|z_i = k, \Sigma_k, \vec{\mu}_k &\sim \text{G}(\cdot|\vec{\mu}_k, \Sigma_k) && (9) \end{aligned}$$

where $\vec{\theta}_j \sim \vec{H}$ stands for:

$$\begin{aligned} \Sigma_k &\sim \text{Inverse Wishart}_{v0}(\Lambda_0) && (10)\\ \vec{\mu}_k &\sim \text{G}(\vec{\mu}_0, \Sigma_k/K_0) && (11) \end{aligned}$$

and $\vec{\pi}|\alpha \sim \text{Stick}(\alpha)$ is a shorthand of:

$$\begin{aligned} \pi'_k|\alpha &\sim Beta(1, \alpha) && (12)\\ \pi_k &= \pi'_k \prod_{l=1}^{K-1}(1 - \pi'_k),\ K \to \infty. && (13) \end{aligned}$$

The Inverse Wishart distribution in (10) is chosen [16], [17] because it is the conjugate prior for the normal distribution, which is instrumental in deriving a closed form solution for the posterior distribution of the indicators, $\vec{Z} = [z_1, \ldots, z_N]$. In FGMM, the distribution of the weights is modeled as the Dirichlet distribution. However, when the number of classes goes to infinity, it is hard to sample π directly from the Dirichlet distribution. Instead, another process, called *the stick breaking construction* [9] is used and defined as follows. We know that $\sum_{i=1}^{K} \pi_k = 1$, hence, we start with a stick of length 1 and break it into two parts at π'_1, sampled according to the Beta distribution (12). Assign the weight π_1 equal to the length of either one of the two parts (13) and repeat the same process on the other part. The same procedure is repeated until a sufficient number of weights are obtained.

Definition 3 fully describes the way that the feature space is generated. The number of wireless devices is unknown and assumed to be infinity initially. Each device creates a cluster of feature points in the feature space. Each cluster has a distribution, which can be represented by a parameter vector $\vec{\theta}_k$. The parameter vector includes two components, $\vec{\mu}_k$ and Σ_k where $\vec{\mu}_k$ can be understood as an aggregate of the means of the phase shift difference and the means of the frequency difference. The same interpretation can be applied for Σ_k. Due to many effects during the production that contribute to the variability of both features and the law of large number, we can assume that $\vec{\mu}_k$ also follows a Gaussian distribution. The mixing weight, $\vec{\pi}$, i.e., the proportion of the whole feature points each device contributed to is created according to the Stick Breaking Process. α represents our confidence on the model. The larger the α, the stronger our confidence in choosing the base distribution, $\vec{H}$ and the more concentrated the distributions of $\vec{\theta}_k$'s around the base distribution.

V. Fingerprinting for Intrusion Detection

In the previous section, we describe the generative model from which our data set, $\vec{X}$, are generated given the model distributions, the hyperparameters and α. Now, we need to resolve the inverse problem, namely, given the data set, $\vec{X}$, the hyperparameters and α, we have to determine the parameters

of the distributions and the indicators to classify the feature points. We start with assuming the number of clusters is infinite but eventually arrive at a finite number. Consequently, based on the clustering results, we propose the fingerprint identification algorithm and discuss how to perform malicious attack detection.

Our goal is to determine the indicators, z_i, $i = 1 \dots N$, for the feature points. For that purpose, we need to derive an expression for the distribution of $\vec{Z}$ given the prior knowledge and then, use the Gibbs sampling method to sample from the distribution and find the class label with the Maximum a Posteriori.

Gibbs sampling is a method to generate samples of two or more random variables from a joint distribution [21]. Sampling from an univariate distribution is easy to implement. However, when there are multiple variables, direct sampling from a complex joint distribution is difficult. The Gibbs sampler was introduced as an efficient method for this situation. In the Gibbs sampler, in each step, all variables are fixed except for one. The new sample of the unknown variables can be obtained based on its marginal distribution and the samples of the known variables. The process is repeated for all variables and proved to converge to the designated joint distribution after a few steps. To apply the Gibbs sampling method to obtain $\vec{Z}$, we need to determine the marginal distribution of each z_i.

From Figure 2(b), we see that the indicator set depends on its parent, $\vec{\pi}$, its child, $\vec{X}$, and its child's parents, $\vec{\theta_k}$ and $\vec{H}$. Therefore, we want to calculate $P(z_i = k|\vec{Z}_{-i}, \alpha, \vec{\theta}, \vec{H}, \vec{X})$. Here, z_i is the unknown variable while $\vec{Z}_{-i}$ is a vector of known variables. By applying the Bayesian rule, the distribution is:

$$P(z_i = k|\vec{Z}_{-i}, \alpha, \vec{\theta}, \vec{H}, \vec{X}) = P(z_i = k|\vec{Z}_{-i}, \alpha, \vec{\theta}_k, \vec{H}, \vec{x}_i) \quad (14)$$
$$\sim P(\vec{x}_i|z_i = k, \vec{Z}_{-i}, \alpha, \vec{\theta}_k, \vec{H})P(z_i = k|\vec{Z}_{-i}, \alpha) \quad (15)$$
$$\sim P(\vec{x}_i|\vec{\theta}_k)P(z_i = k|\vec{Z}_{-i}, \alpha) \quad (16)$$

where the "$\sim$" symbol hides the normalized factor in the Bayesian rule.

In (16), $P(\vec{x}_i|\vec{\theta}_k)$ is the likelihood and simply a Gaussian distribution. The only unknown term is $P(z_i = k|\vec{Z}_{-i}, \alpha)$, which will be determined in two steps. First, starting with a finite K, we apply the FGMM. Second, we explore the limit when K$\rightarrow \infty$ and apply IGMM.

A. Apply FGMM To Solve For The Classification Problem

With a finite K, the feature space is modeled as FGMM with $\vec{M}$ and $P(\pi|\alpha, \vec{M})$ described as in (5). Assume that we have already classified N feature points, and just received a new one, $\vec{x}_{N+1}$. We need to specify the probability of assigning the new one to a new cluster, which has not contribute any feature point so far, $P(z_i \neq z_j, \forall j \neq i|z_{-i}, \alpha)$. Furthermore, we also need to determine $P(z_{N+1} = k|z_{1:N}; \alpha, \vec{M})$, the probability of assigning the new feature point, x_{N+1}, to an existing cluster with parameters $\vec{\theta}_k$:

$$\begin{aligned} & P(z_{N+1} = k|z_{1:N}; \alpha, \vec{M}) \\ = & \int P(z_{N+1} = k|\vec{\pi})P(\vec{\pi}|z_{1:N}; \alpha, \vec{M})d\vec{\pi} \quad (17) \\ = & \int P(z_{N+1} = k|\vec{\pi})P(\vec{\pi}; \alpha^*, \vec{M}^*)d\vec{\pi} \quad (18) \\ = & E(P(z_{N+1} = k|\vec{\pi})) \quad (19) \\ = & E(\pi_k) = \frac{\alpha^* m_k^*}{\sum_{i=1}^K \alpha^* m_i^*} = m_k^* \quad (20) \end{aligned}$$

(17) is simply a marginal distribution where $\vec{\pi}$ is margined out. (19) is derived from (18) according to the definition of the expectation. In (19), $P(z_{N+1} = k|\vec{\pi})$ is actually π_k. As a result, we have the expected probability or the marginal probability of assigning the new feature point to a represented class as in (20). m_k^* belongs to $\vec{M}^*$ and $\vec{M}^* = [m_1^*, ..., m_k^*, ..., m_K^*]$. $\vec{M}^*$. α^* are updated prior parameters of the Dirichlet distribution after observing $z_{1:N}$ feature points [14]. To find out the updated prior parameters, applying the Bayesian rule, we have:

$$P(\vec{\pi}|z_{1:N}; \alpha, \vec{M}) = P(z_{1:N}|\alpha, \vec{M}, \vec{\pi})P(\vec{\pi}|\alpha, \vec{M}) \quad (21)$$
$$= A \prod_{k=1}^K \vec{\pi}_k^{n_k} x \prod_{i=1}^K \vec{\pi}_k^{\alpha m_k - 1} = A \prod_{k=1}^K \vec{\pi}_k^{\alpha m_k + n_k - 1} \quad (22)$$
$$= Dir(\vec{\pi}; \alpha^*, \vec{M}^*) \quad (23)$$

where $m_k = \alpha/K$, n_k is the number of feature points in the same cluster, and A is a normalizing constant. m_k represents the number of feature points that belong to class k^{th} initially. In (21), the distribution of $z_{1:N}$ is just a Multinomial distribution, while $\vec{\pi}$ given $\alpha, \vec{M}$ follows the Dirichlet distribution. Since the Dirichlet distribution is the conjugate prior for the Multinomial distribution, the posterior distribution of the weights is the Dirichlet distribution with the updated prior parameters, α^* and $\vec{M}^*$ as in (23). According to [19], the updated prior parameters can be calculated as:

$$\alpha^* = \alpha + N \text{ and } \vec{M}^* = \frac{\alpha M + N\hat{F}}{\alpha + N}, \quad (24)$$

where $\hat{F}$ is the empirical distribution. Applying the above results, we can find the updated prior parameters in (23):

$$m_k^* = \frac{\alpha m_k + \sum_{i=1}^N \delta(z_i = k)}{\alpha + N} \quad (25)$$

From (23) and (25), we have:

$$\begin{aligned} P(z_{N+1} = k|z_{1:N}, \alpha, \vec{M}) &= \frac{\alpha m_k + \sum_{i=1}^N \delta(z_i = k)}{\alpha + N} \\ &= \frac{\alpha/K + n_k}{\alpha + N}, \quad (26) \end{aligned}$$

where n_k is the number of feature points coming from the same k^{th} cluster excluding the feature point $N + 1$.

B. IGMM

Now, we will explore the mixture by letting $K \rightarrow \infty$. When K approaches infinity, the FGMM becomes an IGMM.

Equation (26) is equivalent to:

$$P(z_{N+1}=k|z_{1:N},\alpha) = \frac{\alpha m_k + \sum_{i=1}^{N}\delta(z_i=k)}{\alpha+N} \approx \frac{n_k}{\alpha+N}. \quad (27)$$

The value on the right hand side of (27) is the probability that the $(N+1)^{th}$ feature point belongs to the k^{th} device. One important property of the NPB is the exchangeability [23]. With this property, we can swap $N+1$ with any i number without changing the joint probability. Hence, we have:

$$P(z_i=k|\vec{Z}_{-i},\alpha) = \frac{n_{k,-i}}{\alpha+N-1}, \quad (28)$$

where $n_{k,-i}$ is the number of observations assigned to the k^{th} cluster, excluding the i^{th} observation. The left hand side of (28) is the probability of assigning a feature point to an existing device. The probability of assigning that feature point to a new device is given by,

$$P(z_i \neq z_j, \forall j \neq i|z_{-i},\alpha) = 1 - \frac{\sum_{j=1}^{K} n_{j,-i}}{\alpha+N-1} \quad (29)$$

$$= \frac{\alpha}{\alpha+N-1}. \quad (30)$$

This probability is simply equal to (1 - sum of all probability of assigning to K existing classes). The summation in the right hand side of (29) is equal to $N-1$ because it counts all the data except for the current one. This is also called the Chinese Restaurant Process (CRP). In CRP, the number of tables (clusters), is infinite and each table can serve an infinite number of customers (feature points). The first customer coming in seats at the first table. The second one will seat either at the first table or a new one. The probability of seating at an occupied table is proportional to the number of customers already seated there in (28). And the probability of seating in a new table is proportional to α in (30).

From (28) and (29), one can easily design a Gibbs sampler to obtain samples of Z. Nevertheless, we see that there is no observed data in the above equations, only priors. After observing $N-1$ feature points, we can update our priors by applying the Bayesian rule. Using (28) and (29) as priors in (16), we can update the posterior of the indicator variables after observing Z_{-i} feature points. The probability of assigning a feature point to a represented class is:

$$P(z_i=k|\vec{Z}_{-i},\alpha,\vec{\theta},\vec{H},\vec{X}) = P(z_i=k|\vec{Z}_{-i},\alpha)P(\vec{x}_i|\vec{\theta}_k) = \frac{n_{k,-i}}{\alpha+N-1}Gaussian(\vec{\theta}_k). \quad (31)$$

The probability of assignment to unrepresented classes is,

$$P \quad (z_i \neq j, \forall j \neq i|\vec{Z}_{-i},\alpha,\vec{\theta},\vec{H},\vec{X}) \quad (32)$$

$$= P(z_i \neq j, \forall j \neq i|\vec{Z}_{-i},\alpha)P(\vec{x}_i;\vec{H})$$

$$= \frac{\alpha}{\alpha+N-1}\int_{\vec{\theta}} P(\vec{x}_i|\vec{\theta})P(\vec{\theta}|\vec{H})d\vec{\theta}.$$

The integration gives the likelihood of being assigned to any unrepresented classes. It is also the marginal probability since $\vec{\theta}$ is integrated out. The integration is analytically tractable because we have chosen $P(\vec{\theta}|H) \sim$ Inverse Wishart$_{v0}(\Lambda_0)$ which is the conjugate prior of the Normal distribution. The above two equations are necessary and sufficient conditions to implement a Gibbs Sampling algorithm. However, in (31) we see that in order to sample the values of Z, we need to sample the values of the parameters in advance. If, somehow, we can integrate out the parameters, the sample space will be much reduced and the algorithm will be faster. We adopt the Collapsed Gibbs Sampling Method [20] to improve the computation efficiency.

C. Collapsed Gibbs Sampling Method

From Figure 2(b), applying the Bayesian rule, we have the joint distribution:

$$P(\vec{Z},\vec{\theta}|\vec{X};\alpha,\vec{H}) = P(\vec{\theta}|\vec{H})P(\vec{Z}|\vec{X},\alpha) \quad (33)$$

$$\sim P(\vec{\theta}|\vec{H})P(\vec{X}|\vec{Z},\vec{\theta};\vec{H})P(\vec{Z}|\alpha). \quad (34)$$

Now, when $z_i=k$, the joint distribution becomes

$$P(z_i=k,\vec{\theta}_k|\vec{X};\alpha,\vec{H}) = P(\vec{\theta}_k;\vec{H})P(\vec{X}_k|\vec{\theta}_k;\vec{H})P(z_i=k;\alpha), \quad (35)$$

where $\vec{X}_k$ is the set of all feature points belonging to the k^{th} cluster. The marginal joint distribution after integrating out the parameters is:

$$P(z_i=k|\vec{X};\alpha,\vec{H}) = \int P(z_i=k,\vec{\theta}_k|\vec{X};\alpha,\vec{H})d\vec{\theta}_k \quad (36)$$

$$= P(z_i=k;\alpha)\times\int P(\vec{X}_k|\vec{\theta}_k;\vec{H})P(\vec{\theta}_k|\vec{H})d\vec{\theta}_k \quad (37)$$

$$= P(z_i=k;\alpha)\times E\{P(\vec{X}_k|\vec{\theta}_k;\vec{H})\} \quad (38)$$

$$= P(z_i=k;\alpha)\times P(\vec{X}_k;\vec{H}). \quad (39)$$

From (36), we apply (35) to get (37), and (38) is derived by applying the definition of expectation value in (37). Given the $P(z_i=k|\vec{X};\alpha,\vec{H})$, we are now able to calculate

$$P(z_i=k|\vec{Z}_{-i},\vec{X};\alpha,\vec{H}) = P(z_i=k|\vec{Z}_{-i};\alpha)\times P(\vec{x}_i|\vec{X}_{k,-i};\vec{H}), \quad (40)$$

where $P(z_i=k|\vec{Z}_{-i};\alpha)$ is given in (31) . Choosing the conjugate priors, according to [17], we will have the multivariate Student-t distribution for $P(\vec{x}_i|\vec{X}_{k,-i};\vec{H})$:

$$P \quad (\vec{x}_i|\vec{X}_{k,-i};\vec{H})$$

$$\sim \quad t_{\upsilon_n-D+1}(\vec{\mu}_n,\Lambda_n(\kappa_n+1)/(\kappa_n(\upsilon_n-D+1))), \quad (41)$$

where

$$\vec{\mu}_n = \frac{\kappa_0}{\kappa_0+N}\vec{\mu}_0 + \frac{N}{\kappa_0+N}\bar{X}, \quad (42)$$

$$\kappa_n = \kappa_0+N,\ \nu_n=\nu_0+N,$$

$$\Lambda_n = \Lambda_0 + S + \frac{\kappa_0 n}{\kappa_0+N}(\bar{X}-\vec{\mu}_0)(\bar{X}-\vec{\mu}_0)^T,$$

$$\bar{X} = (\vec{x}_1+\vec{x}_2+\cdots+\vec{x}_N)/N, \quad (43)$$

where D is the data dimension, $\vec{\mu}_n,\kappa_n,\nu_n$ and Λ_n are updated hyperparameters after observing $N-1$ samples. To sample $\vec{Z}$, we do not have to sample the parameters. Instead we can implement the Collapsed Gibbs Sampling. For the case of assignment to an unrepresented cluster, since in (32), the parameters are already integrated out, we only need to find

$P(\vec{x}_i; \vec{H})$. Based on the expression of $P(\vec{x}_i|\vec{X}_{k,-i}; \vec{H})$ given in (41), the distribution can be determined as a multivariate Student-t distribution with the hyperparameters before updating, $\vec{\mu}_0, \kappa_0, \nu_0$ and Λ_0:

$$P(x_i; \vec{H}) \sim t_{v_0-D+1}\left(\vec{\mu}_0, \Lambda_0 \frac{\kappa_0+1}{\kappa_0(v_0-D+1)}\right). \quad (44)$$

In conclusion, we have obtained two posterior distributions for the indicators. The first distribution is used in the case of assigning the feature point to an existing cluster. From (28), (41) and (40) we have:

$$\begin{aligned} & P(z_i = k|Z_{-i}, X; \alpha, \vec{H}) \quad (45) \\ \sim \quad & \frac{n_{k,-i}}{\alpha+N-1} t_{v_n-D+1}\left(\vec{\mu}_n, \Lambda_n \frac{\kappa_n+1}{\kappa_n(v_n-D+1))}\right). \end{aligned}$$

The second distribution is used in the case of assignment to a new device, which has yet to contribute any data point in the feature space. From (30), (44) and (40), we have:

$$\begin{aligned} P \quad & (z_i \neq j, \forall j \neq i|Z_{-i}, X; \alpha, \vec{H}) \quad (46) \\ \sim \quad & \frac{\alpha}{\alpha+N-1} t_{v_0-D+1}\left(\vec{\mu}_0, \Lambda_0 \frac{\kappa_0+1}{\kappa_0(v_0-D+1)}\right). \end{aligned}$$

The posterior allows the application of a Gibbs sampler to sample the values for the indicators. Below is the Collapsed Gibbs sampler algorithm [16] to classify the feature points into different classes. In each step, each indicator will be updated with a new value sampled from either (45) or (46). After a number of steps, the samples obtained will converge to their original joint distribution.

Algorithm 1: Collapsed Gibbs Sampler:

Create random integer samples for the indicators in the range from $1:N$. Start with $K = max(\vec{Z})$.

- At j^{th} step
  - Update the indicator for observation 1.
    - Check how many observations have the same indicator $z_1^{(j)}$. If only x_1, update the number of classes: $K = K-1$ and $z_2^{(j-1)} = z_2^{(j-1)} - 1, \ldots, z_N^{(j-1)} = z_N^{(j-1)} - 1$.
    - Sample $z_1^{(j)}$ according to Equations (45) and (46), given $z_2^{(j-1)}$,$z_3^{(j-1)}$,...,$z_N^{(j-1)}$.
    - If $z_1^{(j)} > K$, update $K = K+1$.
  - Update the indicator for observation 2.
    - Check how many observations have the same indicator $z_2^{(j)}$. If only x_2, update the number of classes: $K = K-1$ and $z_3^{(j-1)} = z_3^{(j-1)} - 1, \ldots, z_N^{(j-1)} = z_N^{(j-1)} - 1$.
    - Sample $z_2^{(j)}$ according to Equations (45) and (46), given $z_1^{(j)}$,$z_3^{(j-1)}$,..., $z_N^{(j-1)}$.
    - If $z_1^{(j)} > K$, update $K = K+1$.

    ⋮
  - Update the indicator for observation N.
    - Check how many observations have the same indicator $z_N^{(j)}$. If only x_N, update the number of classes: $K = K-1$.
    - Sample $z_N^{(j)}$ according to Equations (45) and (46), given $z_1^{(j)}$,$z_2^{(j)}$,...,$z_{N-1}^{(j)}$.
    - If $z_N^{(j)} > K$, update $K = K+1$.

In the Gibbs sampler, as soon as a new indicator is sampled from the posterior, it will be used to update the hyperparameters to sample the next indicator.

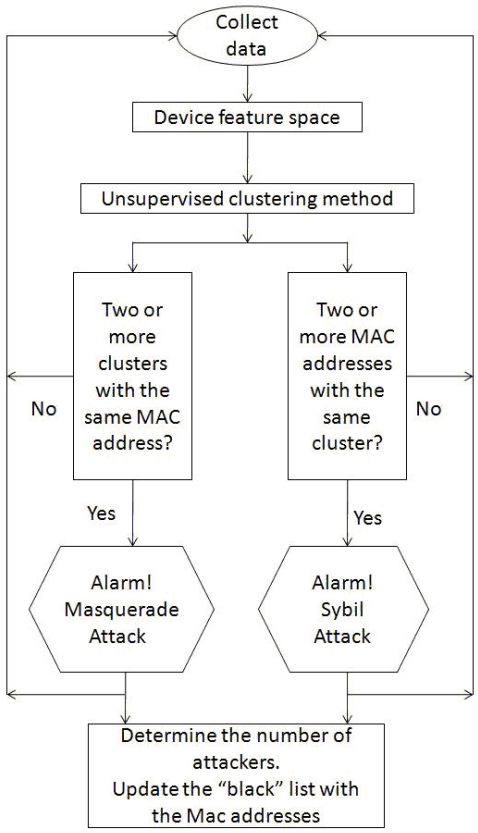


Fig. 3. Attack Detection Scheme

D. Attack Detection Scheme

Finally, Figure 3 summarizes the attack detection scheme applying the afore-mentioned classification technique. The technique is unsupervised since there is no training phase. Signals are measured by a sensing station continuously to detect if there is any attack. Under this scenario, two types of attacks can be detected, Sybil attack and masquerade attack. From the results of the unsupervised clustering method, if two or more MAC addresses are associated with the same cluster, it means the system is under the Sybil attack in which one device assumes multiple identities. On the other hand, if we detect two or more clusters with the same MAC address, it means the system is under the masquerade attack (spoofing) in which multiple devices try to share the same identity. We are also able to determine the number of attackers and update the "black" list for those malicious nodes with the MAC addresses.

VI. Simulation Result

To evaluate the performance of our proposed algorithm implemented in Matlab, we consider three metrics:

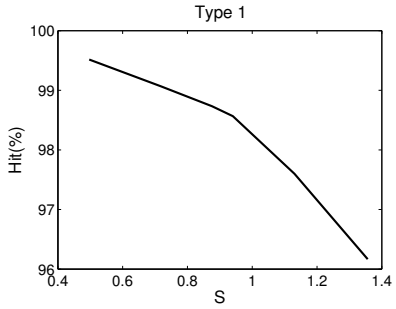


Fig. 4. Masquerade attack detection probability (type 1)

1) Type 1: the hit rate of detecting the correct number of clusters, defined as the number of trials with correct count over the total number of trials.
2) Type 2: the hit rate of assigning every single feature point to its correct cluster, defined as the number of feature points assigned to their original clusters over the total number of feature points.
3) Type 3: the false alarm rate of assigning a feature point to a valid cluster other than the correct one.

Note the last two metrics differ if spurious clusters are generated in the clustering process. In the simulation, we follow the generative models in Section IV, and vary the parameter sets. To characterize the difference and similarity between any two distributions. $\vec{\theta}_1$ and $\vec{\theta}_2$, we define

$$Diff = \frac{|\mu_1 - \mu_2|}{3\sigma_1 + 3\sigma_2} \text{ and } S = \frac{1}{Diff}$$

where $Diff$ and S are the difference and similarity, respectively, $\mu_1(\mu_2)$ and $\sigma_1(\sigma_2)$ are mean and standard deviation of distribution 1(2). 3σ is chosen because it accounts for 99.7% of the population of the feature set. When $Diff = 1$, the two clusters are contiguous to each other.

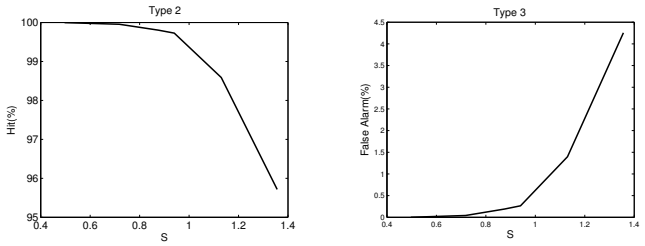
A. Two Devices in a Masquerade Attack

In this scenario, there are two physical devices, one is legitimate and the other is malicious. The malicious device spoofs the MAC address of the legitimate device. In the simulation, the means and the variances of the distributions are chosen to create a range of the similarity, S, from 0.5 to 1.37. In this case, the hit rate of detecting the correct number of attackers is the same as the hit rate of correctly detecting if there is a masquerade attack.

As shown in Figure 4, when the clusters are adjacent to each other ($S = 1$), the probability of detecting the attack is 98.2%. In Figure VI-A, the type 2 hit rate and the false alarm rate also given. As the similarity increases, the hit rate drops and the false alarm rate increases. We can see that the proposed scheme can detect the masquerade attack with high probabilities.

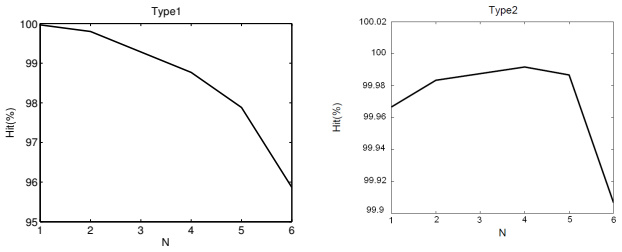
B. Varying Number of Devices

In this set of experiments, the number of devices varies from 1 to 4. In each trial, centers of the clusters corresponding to each device are chosen randomly in an area of 40x40 in the 2-D feature space, with variances of the clusters chosen randomly in the range from 0 to 1. As the cluster centers



(a) Hit rate, type 2 (b) False alarm, type 3

Fig. 5. Correct classification probabilities



(a) Type 1 (b) Hit rate, type 2

Fig. 6. Performance with varying N

are chosen randomly, it is possible that there are multiple clusters centered around the same position. Hence, as in Figure 6(a), when the number of devices increases, the probability of correct detection of the number of clusters decreases. In the situation where there is a masquerade attack, this probability is effectively the probability of correctly counting the number of attackers. For example, if there are only two legitimate devices and one malicious device spoofing the MAC address of them, our method can detect ID spoofing with probability 99.3% (corresponding to the case $N = 3$ in Figure 6(a)). In the situation where one attacker tries to emulate multiple devices by declaring multiple MAC addresses to implement a Resource Depletion Attacks [2] or Sybil attack, our algorithm have a hit ratio of 99.8% in Figure 6(a). This is accomplished by correlating the MAC addresses with the cluster labels. If observations from the same cluster are associated with different MAC addresses, such attacks can be detected.

C. Experimental Results

To test the effectiveness of our proposed method, we set up the following experiment: 4 Zigbee boards are used as the transmitting devices, and a USRP2 board is used as the detecting station. By using the CSMA mechanism in the MAC layer, 4 Zigbee boards can communicate wirelessly while not interfering each other. The detecting station keeps track of radio signals. Our objective is to investigate the situation where the channel has a non-negligible affect on the received signal and may cause some variations in the fingerprints. All the Zigbee boards are programmed with the same parameters, e.g., 20 bytes MAC payload for each packet, 2MHz bandwidth, 2.48GHz center frequency (channel 26), and 0dBm transmitted power. On the receiving USRP2 board, the signal is first

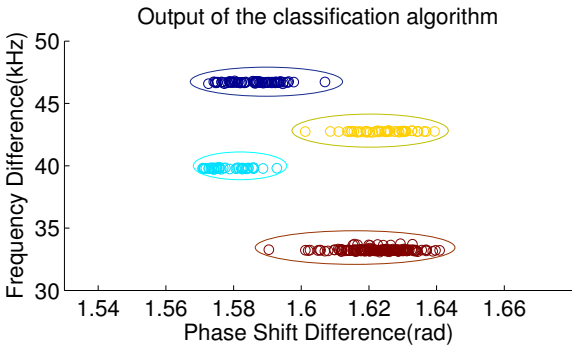


Fig. 7. Classified Feature Space

down-converted from 2.48GHz and decimated from 100 Mega samples per second (Msps) to 4Msps. After passing through the Root Raised Cosine Filter (RRCF), the signal can be observed within the interest range. Then, it is synchronized to compensate the carrier offset and phase difference by clock recovery and PLL (phase-locked loop). After passing through the low pass filter, we receive the OQPSK signal. By averaging over one frame, we can collect frequency difference and phase shift difference for each device, forming the feature space as the input to the Attack Detection Scheme in Figure 3.

Even with background noise and channel effects, our algorithm is shown to perform well, with type 1 and type 2 hit rates at 100% and type 3 error at 0%. When the number of devices increases from 1 to 4, the performance does not degrade. Figure 7 shows the output of the classification algorithm from the this experiment. Each circle corresponds to one device and is displayed in different colors. All the feature points from same device locate inside their respective circles. Since each device has a distinctive feature cluster location, hence, they can be effectively classified by the proposed algorithm.

VII. Conclusion

In this paper, we employed a nonparametric Bayesian approach to identify wireless devices by their transmitter characteristic, so as to prevent the attacks such as Sybil attacks and masquerade attacks. The Infinite Gaussian Mixture Model was utilized for modeling, and a collapsed Gibbs sampling method was constructed for device identification. From the simulation and experimental results, the proposed scheme demonstrated superior performance in detecting these attacks as well as identifying malicious nodes.

References

[1] J. R. Douceur, "The Sybil Attack" in proceedings of *the 1st International workshop on Peer-To-Peer Systems*, Cambridge, MA, Mar 2002.

[2] D. B. Faria and D. R. Cheriton, "Detecting IdentityBased Attacks in Wireless Networks Using Signalprints", in proceedings of *the 5th ACM Workshop on Wireless Security*, Los Angeles, California, USA, 2006.

[3] T. M. Gil and M. Poletto, "MULTOPS: A Data-structure for Bandwidth Attack Detection", in the proceedings of *the 10th Usenix Security Symposium*, Washington, D.C, USA, August 2001

[4] D. Moore, G. Voelker and S. Savage, "Inferring Internet Denial of Service Activity", *ACM Transactions on Computer Systems (TOCS)*, Volume 24 , Issue 2, pp:115-139, May 2006.

[5] K. Xing and X. Cheng, "From Time Domain to Space Domain: Detecting Replica Attacks in Mobile Ad Hoc Networks", in proceedings of *the 29th Conference on Information Communications*, San Diego, California, USA, March 2010.

[6] S. Bratus, C. Cornelius, D. Kotz, and D. Peebles"Active behavioral Fingerprinting of Wireless Devices", in proceedings of *the 1st ACM Conference on Wireless Network Security*, Alexandria, VA, USA, March 2008.

[7] J. Hall, M. Barbeau, and E. Kranakis, "Radio frequency fingerprinting for Intrusion Detection in Wireless Networks", DRAFT, 2005, [online] Available: $http : //people.scs.carleton.ca/ \ kranakis/Papers/IDSRFFv4 - 4.pdf$

[8] 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*, San Francisco, California, USA, September 2008.

[9] Y. W. Teh, *Dirichlet Processes*, Encyclopedia of Machine Learning, Springer, New York, 2007.

[10] A. Candore, O. Kocabas, and F. Koushanfar, "Robust Stable Radiometric Fingerprinting for Wireless Devices", in the proceedings of *IEEE International Workshop on Hardware-Oriented Security and Trust, HOST '09*, San Francisco, CA, USA, July 2009.

[11] Ettus Research LLC, [online], Available: $http : //www.ettus.com/$.

[12] J. Yang, Y. Chen, W. Trappe, and J. Cheng, "Determining the Number of Attackers and Localizing Multiple Adversaries in Wireless Spoofing Attacks", in the proceedings of *the 28th Conference on Computer Communications*, Rio de Janeiro, Brazil, April 2009.

[13] Y. W. Teh, "Dirichlet Process, Tutorial and Practical Course", MLSS 2007, [online] Avaialbe: $http : //www.gatsby.ucl.ac.uk/ ywteh/research/npbayes/mlss2007.pdf$

[14] C. E. Rasmussen, "The Infinite Gaussian Mixture Model", in the proceedings of *Neural Information Processing Systems (NIPS) 2000*, Denver, CO, USA, 2000.

[15] P. W. Diaconis, M. L. Eaton and S. L. Lauritzen, "Finite De Finetti Theorems in Linear Models and Multivariate Analysis", *Scandinavian Journal of Statistics*, Vol. 19, No. 4, pp:289-315, 1992.

[16] F. Wood and M. J. Black, "A Nonparametric Bayesian Alternative to Spike Sorting", *Journal of Neuroscience Methods*, pp 173(1):1-12, Aug. 2008.

[17] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, Chapman & Hall/CRC, London, 2nd edition, 2003.

[18] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet Processes", *Journal of the American Statistical Association*, JASA, pp: 101(476):1566-1581, 2006.

[19] A. Ranganathan, *the Dirichlet Process Mixture Model*, DRAFT, 2004 [Online] Available: $http : //biocomp.bioen.uiuc.edu/journal_club_web/dirichlet.pdf$.

[20] J. S. Liu, "The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem", *Journal of the American Statistical Association*, Vol. 89, No. 427, pp:958-966, Sep., 1994.

[21] P. Resnik and E. Hardisty, *Gibbs Sampling for the Uninitiated*, Technical report, UMIACS, 2009.

[22] N. Bouguila and D. Ziou, "A Dirichlet Process Mixture of Generalized Dirichlet Distributions for Proportional Data Modeling", *IEEE Transactions on Neural Networks*, Volume 21 , Issue 1, pp 21(1):107-122, January 2010.

[23] Z. Ghahramani, "Nonparametric Bayesian methods", *Tutorial presentation at the UAI Conference*, 2005.

[24] E. Jackson, M. Day, A. Doucer, and W. J. Fitzgerald, "Bayesian Unsupervised Signal Classification by Dirichlet Process Mixtures of Gaussian Processes", in the proceedings of *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007*, Honolulu, Hawai'i, U.S.A., April 2007.

[25] M. D. Escobar and M. West, "Bayesian Density Estimation and Inference Using Mixtures", *Journal of the American Statistical Association*, Vol. 90, No. 430, pp. 577-588, Jun., 1995.