Detecting Impersonation Attacks in a Static WSN

VIKTOR SELLEBY
ANTON LIN
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Detecting Impersonation Attacks in a Static WSN

Viktor Selleby, Anton Lin

tna12vse@student.lu.se, dic13ali@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisors: Martin Hell, LTH
Simon Johansson, Tritech Technology

Examiner: Thomas Johansson

4th October 2017

Abstract

The current state of security found in the IoT domain is highly flawed, a major problem being that the cryptographic keys used for authentication can be easily extracted and thus enable a myriad of impersonation attacks. In this MSc thesis a study is done of an authentication mechanism called device fingerprinting. It is a mechanism which can derive the identity of a device without relying on device identity credentials and thus detect credential-based impersonation attacks.

A proof of concept has been produced to showcase how a fingerprinting system can be designed to function in a resource constrained IoT environment. A novel approach has been taken where several fingerprinting techniques have been combined through machine learning to improve the system's ability to deduce the identity of a device.

The proof of concept yields high performant results, indicating that fingerprinting techniques are a viable approach to achieve security in an IoT system.

Keywords — IoT, Device Fingerprinting, Machine Learning, Machine to Machine Authentication, Impersonation Attacks, Spoofing Attacks, Wireless Sensor Network

i

Acknowledgements

Popular Science Summary

Two devices that communicate without presence of a human being usually identify themselves using cryptographic keys stored in their firmware. Studies have shown that it is possible to extract such keys from the devices and thus impersonate them. The proposed solution to this problem is to add an additional layer of security to identify such impersonations.

There exists several methods to authenticate a device that are not based on cryptographic keys alone. One such method is physical device fingerprinting which stems from the fact that physical properties of a device contains some measure of uniqueness. Thus, these physical properties can be used to authenticate a device.

Studies have previously been done to test the accuracy of such methods, however, most have been tested in laboratory environments with high-end equipment. As a large number of the systems that communicate without human presence are low-cost and low-power it was unclear if physical device fingerprinting would be possible in such systems.

Two of the physical device fingerprinting methods analyzed were deemed possible to implement, clock skew and RSSI mean clustering. Clock skew is based on minuscule differences in devices' perception of time and RSSI mean clustering utilizes many antennae to see if a device moves to a different location.

Authentication tests using both of the implemented fingerprinting methods and four tested devices showed very good results. The system demonstrated that it is possible to detect an impersonation attempt in all the tests and very rarely misjudged a legitimate device as an impersonating device. However, the system has to be tested more extensively with an increasing number of tested devices as well as in different environmental settings to see if the results hold.

Contents

List of Figures

x

List of Tables

List of Acronyms

- **AP** — Access Point
- **BLE** — Bluetooth Low Energy
- **IC** — Integrated Circuits
- **IoT** — Internet of Things
- **LAN** — Local Area Network
- **MAC** — Media Access Control
- **M2M** — Machine to Machine
- **PDF** — Probability Distribution Function
- **PKI** — Public Key Infrastructure
- **PUF** — Physically Uncloneable Function
- **RF-DNA** — Radio Frequency Distinct Native Attribute
- **RFF** — Radio Frequency Fingerprint
- **RFID** — Radio Frequency Identification
- **RSSI** — Received Signal Strength Indication
- **SNR** — Signal to Noise Ratio
- **WLAN** — Wireless LAN
- **WSN** — Wireless Sensor Network

Introduction

With the rise of Internet of Things (IoT), the number of embedded devices with Internet connectivity is seeing an exponential growth. A society that is fully connected down to the bare things in our everyday life has many benefits but does also raise a number of security concerns, autonomous authentication between machines –machine to machine authentication– being one of them.

Most IoT systems employ credential based authentication. Similar to smart hand-held devices or traditional mobile phones, it is the keys stored in the device's non-volatile memory that are the very credentials on which the devices are authenticated. However, unlike smart hand-held devices, IoT devices do not enjoy the security of being carried on the owner's physical being, they are most often unmanned and unmonitored. Thus, IoT devices are highly susceptible to attacks where a potential adversary could extract the necessary credentials and pose as the device in question.

1.1 Background

Authentication of embedded devices are in many cases credential based which can be seen in [1] and [2]. A device is deemed authenticated by being able to produce the credentials, most often stored in a SIM-card or the device's firmware, during the authentication procedure. Credentials in this context denotes the device's PKI keys. The type of authentication utilized is called machine to machine (M2M) authentication, where two machines interact with each other without the presence of a human being.

In traditional use cases such as hand held smart devices or personal computers, human presence is a fact. In such cases, an assumption can be made that the devices are not compromised since they are located at the owner's physical being. Thus a potential adversary does not have the physical presence necessary to extract said credentials from the devices in question. However, what means are available to assert that a physical device is not in fact compromised when the owner of said device is not physically present?

Experiments done in [3] showed that it is possible to retrieve the PKI keys by

reading the flash of a node without expensive hardware or software. Thus, the paper emphasizes the need to secure such nodes. With the PKI keys, an adversary can then impersonate a node as presence of the keys often is the only means of authentication when initiating a connection [4].

Research has been done on various authentication mechanisms that operate on top of the traditional credential based authentication. In machine-metrics based authentication various metrics such as computation time given a certain random problem, hardware fingerprint and other characteristics can be used to deduce the identity of the authenticating device. The core of the concept is to identify the proving party through information which cannot be readily found in the device's non-volatile memory. Such mechanisms are often designed for more traditional systems such as personal computers or smart hand held devices. However, can such authentication models be extended to embedded systems where computational resources are more restrained?

The target environment of this project is Tritech's demonstration system consisting of end devices in the form of temperature sensors which are equipped with system on a chip Mighty Gecko [5], the sensors comprise a fog network whose gateway is realized by a Raspberry Pi with Internet connectivity which in turn is communicating with an Amazon Web Services (AWS) deployment. The AWS instance enables an application interface which can be reached from desktops or smart hand-held devices to read data from the different end devices but also to control them. The design of the solution to be produced in the project will be specifically aimed towards systems with similar characteristics to the ones found in Tritech's demonstration platform.

1.2   Project Aims

The main goal of the Master's thesis is to investigate different means to continuously authenticate a device. This will be implemented in a monitoring system designed to detect compromised nodes in an IoT network. Thus, the questions which this thesis will try to answer are the following:

1. Given a node in an embedded systems network whose private credentials have been compromised, what mechanisms can be deployed to identify and subsequently reject data originating from such a node?

    (a) The methods for authentication which will be evaluated will fall within the category of machine-metrics based authentication, the core of the concept is to identify the proving party through information which cannot be readily found in the device's non-volatile memory.

    (b) Due to the resource constrained nature of the environment, an analysis will be done on the hardware and software requirements of the implemented system and how the resource demands scale in a network of many nodes.

2. From the analysis in 1, can a system be produced and tailored to fit onto a resource constrained environment? What compromises will need to be done in order to achieve such an implementation and what impact will such compromises have on the security level of the system?

3. Can the implementation produced distinguish between a legitimate and corrupt node regardless of possession of underlying credentials?

1.3 Scope

The scope of the MSc thesis is to investigate authentication techniques which are operating on top of credential based authentication and apply these techniques to an embedded wireless system.

System Characteristics

The system in question will be a static one, nodes are fixed and will not change their spatial location without an explicit reconfiguration procedure where the new location of the node in question is registered. The network topology is star-shaped with the access point acting as the center of the topology. The network hop count is limited to one hop, meaning that each node will have to be directly connected to the AP. In order to conduct a well-defined discussion on the topic, a thorough definition of terms used throughout the report is given.

- **Credentials**, data found in device memory which the device can use to prove its identity. E.g. MAC address and cryptographic keys.

- **Embedded system**, a system whose computational abilities are highly constrained in comparison to a more typical system where storage, CPU power and power supply is not considered a highly limited resource. An example of an embedded systems is a wireless sensor network (WSN), where the connected sensors are constrained in their functional and computational capabilities.

- **Authentication**, the act of verifying the identity of a connecting party. An authentication procedure involves two parties: the proving party and the verifying party. The former is in the context of the thesis a connecting device whose goal is to establish its identity. The latter's function is to verify the identity of the proving party and subsequently accept or deny connection based on the outcome of the verification.

- **Compromised**, a device's credentials are said to be compromised if an adversary were to gain unlawful access to them.

1.3.1 Attack model

As mentioned in the background section, the thesis project will evaluate vulnerabilities in an embedded system where the credentials of one or more nodes have been compromised. Once the credentials have been compromised, they can be

relocated to a device at the disposal of the attacker who now has the means to pose as a compromised node. Consider Alice a proving node, Bob a verifying party and Eve a malicious entity. In an authentication scheme where private public key pair constitute the means of authentication, Eve has gained access to the crypto keys used between Alice and Bob. Being in posession of the crypto keys, Eve can now impersonate Alice and send a message to Bob without Bob detecting it. In this project, a protocol will be created such that Bob should be able to detect if a packet originates from Alice or Eve, even if Eve in some way claims to be Alice.

1.4   Outline

The report will be structured as follows: Chapter 2 will present and describe basic concepts to provide preliminary knowledge required to understand following chapters. Chapter 3 intends to describe the fingerprinting techniques discovered during the literature-study as well as methods to generate and authenticate a fingerprint. In Chapter 4, the attacks relevant to the thesis project are presented, including attacks on the different fingerprinting techniques. An evaluation of the fingerprinting techniques based on the previous chapters will be described in Chapter 5 which will result in the techniques to pursue during implementation. In Chapter 6, the experimental setup and methodology will be presented. The results from the experimentation will then be displayed in Chapter 7. Chapters 8 and 9 will deal with discussion of results and conclusion respectively.

Preliminaries

This chapter will describe concepts relevant to the MSc project.

2.1 Wireless Sensor Network

A wireless sensor network (WSN) is a network of connected nodes that most often have limited processing, computing and power resources. These nodes can be equipped with one or more sensors that in some way measure the environment [6]. The number of nodes in a network range from a handful up to thousands. Once deployed, the network is left unattended and the nodes can either be static or mobile depending on their functionality and purpose [6].

2.1.1 Transport Media

The transport media used in a wireless sensor network can vary depending on the application and environment it is used in. Notable transport media in this context include 802.11, Bluetooth Low Energy and 802.15.4 [7].

IEEE 802.11

IEEE 802.11 is a collection of standards for Wireless Local Area Networks (WLAN), the most prominent one being Wifi. Usage of 802.11 is most often seen in local networks where devices communicate on either 2.4GHz or 5GHz frequency bands. With a recently standardised specification [8] frequency band at sub 1GHz were enabled. This makes 802.11 more appropriate to use in constrained environments since it provides larger wireless coverage, lower obstruction losses as well as smaller energy consumption [9].

Bluetooth Low Energy

Bluetooth Low Energy (BLE) is an adaptation of Bluetooth that is designed for ultra-low-power applications. This is achieved by reducing the number of channels, data throughput, lower range among others [10].

BLE defines two distinct device roles: central and peripheral. A central can manage multiple connections with different peripheral simultaneously while a peripheral can only be connected to one central. In context of WSN, an end node will most often act as a peripheral. These can spend idle periods in sleep mode to preserve energy [11].

IEEE 802.15.4

The IEEE 802.15.4 standard [12] which technologies like ZigBee [13] and Thread [14] are based on. It is designed to provide communication where throughput is small in order to enable inexpensive devices with constrained resources to be part of a wireless networks without rapidly depleting their limited power supply.

Received Signal Strength Indicator

Received Signal Strength Indicator (RSSI) [15] is a measure of the perceived power of a received radio signal. It is expressed in decibels and takes on a negative value. The RSSI is perceived as stronger as it grows towards zero.

Signal to Noise Ratio

The Signal to Noise Ratio (SNR) [16] is a measure of how much power a signal has relative the noise on the channel as described by (2.1).

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{2.1}$$

As nodes in WSN often are low-power devices with low transmission power, a noisy channel can affect bit error rate and frame error rate negatively.

2.1.2   Network Topology

To be able to act upon the sensor data generated by each node, the nodes has to be able to communicate with a base station. How the communication between a node and a base station is done is decided by the topology of the network. Commonly used network topologies in WSN is star, multi-hop mesh and two-tier cluster [17] which can be seen in figure 2.1.

The star topology is based on the idea that every node in the network has a direct communication link with the base station. The two-tier cluster topology can be viewed as an expansion of the star topology. Each node communicates with a cluster head that in turn communicate with the base station. Thus, the base station can communicate with many cluster heads which in turn can communicate with many nodes [18]. In the multi-hop topology every node can relay messages from other nodes, hence all nodes do not need to have a direct communication link with the base station [17].

(a) Star topology

(b) Mesh topology

☐ – Cluster Head

● – Base Station

▲ – End node

(c) Two-tier cluster topology
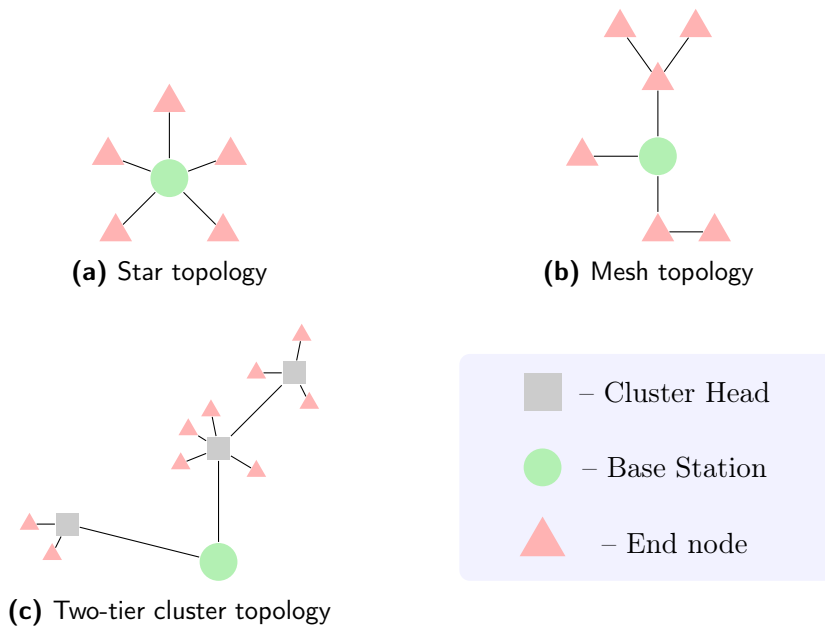
Figure 2.1: Visualisation of different network topologies commonly used in WSN

2.1.3 Applications

WSN offers a lot of application areas including military, environmental, health, home and other commercial applications [19]. Applications that are static by nature and, thus, more relevant to this paper include:

- Home automation applications where sensors are part of domestic devices such as intrusion detection systems, smart meters, household appliances and more.

- Environmental control in buildings as well as in smart agriculture.

- Environmental irregularity detection. Can be used for anything from nuclear launch detection systems to early forest fire detection.

- Industrial machinery.

2.2 Machine Learning

Machine learning is a field where a computer can be trained to make statistical predictions of unseen data based on previously recorded data. Many different machine learning algorithms exist and they can all be divided into two distinct categories, supervised and unsupervised.

2.2.1 Supervised Learning

In supervised learning, the algorithm requires a dedicated training set. It consists of input and its corresponding output where the latter is labeled as belonging to a certain class. The algorithm then uses the labeled data set to find a mapping between input and output that can be used to predict the output of unseen input.

2.2.2 Unsupervised Learning

Unsupervised learning does, in contrary to supervised learning, not require any labeled data set. Instead, the algorithm is trained with only input data. The generated output then depends on the algorithm and its application. In an anomaly detection application using the Isolation Forest algorithm [20], the algorithm will give a verdict on whether or not unseen data is reminiscent of the input data observed during training. A negative verdict will be given if a recorded value is abnormal considered the trained model and a positive one will be given if the data is considered to be in coherence with those in the training data [21].

2.3 F-Score

To measure the accuracy of a model, F-Scores can be used [22]. F-Score is calculated with the rate of true negatives and true positives.

True positives is a measurement of how accurately a model generates positive outcomes given that the outcome is correctly assessed as positive. The true positive score is always a value between 0 and 1. By taking 1 minus the true positive rate, the false negative rate is given. A false negative value is given when a positive outcome is misjudged as a negative. A true negative is given when a negative outcome is correctly judged as a negative and on the contrary, a false positive is given when a negative outcome is misjudged as a positive.

Two measures can be derived from true positives and true negatives: $Precision$ which denotes how accurate a model is in detecting negatives and $recall$ which indicates the model's accuracy in regards to positives. The equation for precision and recall can be seen in (2.2) and (2.3) where FP is false positives, FN is false negatives and TP is true positives.

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

With the precision and recall, the F-score can be calculated as in (2.4). As can be seen by the equation, F-score is a combined measure of precision and recall that ranges from 0 to 1. Because of this, the F-score can be described as a measure of the accuracy of a model.

$$F = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.4}$$

Device Fingerprinting Techniques

In this chapter, techniques to extract device specific attributes – device fingerprints – in order to uniquely identify a device will be described. An evaluation of the various techniques will be done in Chapter 5.

Device fingerprinting techniques can be divided into two categories, active and passive fingerprinting [4]. The former refers to the fact that the verifying party performs some form of active interrogation, causing the proving party to actively respond to said activity. It is the interrogation data on which the verifying party will build the fingerprint. Conversely, passive techniques does not require an interrogation phase to collect data from the verifying party. Data is gathered by monitoring network communications between the two parties, from this a device fingerprint is achieved.

By putting as much logic and computation at the verifying party, the attack surface of the system is limited. For this reason, passive fingerprinting techniques are the main focus of this thesis.

3.1 Fingerprinting Flow

Fingerprinting techniques can vary significantly in their approach and core fundamentals. However, on a higher level, most fingerprinting techniques follow the same steps in their respective procedures [23] [24] [25].

1. **Learning Phase**, a reference fingerprint is generated through data from a legitimate device and stored in the verifying device. This is then used during device authentication, where a derived fingerprint is compared with its corresponding reference fingerprint.

   (a) **Data collection**. The data collection during the learning phase is extensive. A large data set is collected in order to produce a reference fingerprint that is representative for the device. If a small data collection is used to create the reference fingerprint, the effect of outliers and irregular data will have a greater impact.

   (b) **Fingerprint Generation**. During this phase, fingerprinting techniques most often use the collected data to create a reference finger-

print with the use of statistical methods, which vary between finger-
print techniques.

2. **Authentication**. Once a reference database has been created for all au-
thorized devices, the system is open for connecting nodes.

   (a) **Data collection**. The data set collected is significantly smaller than
   the one collected during the learning phase. This, in order to keep
   authentication time to a minimum.

   (b) **Fingerprint Generation**. The fingerprint generation is done in the
   same manner as during the learning phase.

   (c) **Similarity Measurement**. In order to authenticate a device, the
   device's fingerprint has to be matched against the stored reference
   fingerprint for said device. Since fingerprints are estimations, an ex-
   act match is unlikely to be derived. Instead, a threshold is set which
   dictates if the authentication fingerprint is considered a match against
   the corresponding reference fingerprint. Therefore, similarity measure-
   ment techniques are used not only to find similarity metric between
   a derived fingerprint and its corresponding reference, but also to find
   a threshold value where the false negative and false positive rate are
   minimized.

3.2   Clock skew

A computer clock is driven by a quartz crystal. Due to sub-micron imperfections
found within said quartz crystals, each computer's perception of time will differ.
This holds for computers of identical component setup, with components of same
model produced from one single manufacturing batch. Thus, a slight skew in time
will appear between two computers if time is measured with fine enough granu-
larity. The differing perceptions of time is referred to as clock skew [23]and it is
unlikely to be the same for any given pair of computers. A fingerprinting technique
has been developed from this phenomenon and is based on deriving the proving
party's clock skew relative the verifying party's clock.

Research have shown various ways [26] [27] to identify a node through clock skew.
The core concept is to collect data points over time, the points representing time of
departure and time of arrival for a network packet. In [23], the timestamps found
within TCP packets denotes the time of departure of the packet and the time of
arrival is recorded at the receiving end as soon as the packet arrives. When a data
set has been collected, statistical methods are applied to fit a function to the time
series data. The derivative of said function constitutes the clock skew and thus it
is ultimately the function's derivative which dictates the identity of the proving
party.

3.2.1 Data collection

Although numerous clock skew techniques exist, they are all dependent on time of departure and time of arrival in various forms. The authors of [23] have made use of the timestamps found in TCP packets or ICMP packets. In [26] [28], the authors utilize management mechanisms found in the IEEE 802.11 standard to derive timestamps of high granularity from beacon probes. The beacon probes are used to advertise the existence of an Access Point (AP) and synchronize the clock of the connecting party with the AP's clock. 802.11 beacon frames are sent between 10 to 100 times per second which in comparison to regular TCP traffic proves to be a source which will generate a large amount of data in a short amount of time. However, beacon frames are advertised by APs –not devices– which renders the data collection approach in [26] useless in the context of this project.

3.2.2 Fingerprint Generation

The basis of generating a clock skew fingerprint is to calculate the offset in time which the proving device p has in relation to the verifying device v. In [29] the offset o for a specific packet i is calculated by $o_i = v_i^t - p_i^t$ where v_i^t is the time when v receives packet i and p_i^t is the time p sends packet i. The offset is calculated for j packets. The clock skew can then be calculated with linear regression as the slope between $\sum_1^j o_i$ and $\sum_1^j v_i^t$ where v^t constitutes the x-axis and o the y-axis. The accuracy of the clock skew is influenced by the total sample size used in the linear regression as well as the number of outliers in the sample [29].

3.2.3 Similarity Measurement

As stated in Section 3.1, fingerprints are estimations rather than absolute truths. However, the similarity measurement dictates how precise the estimations will be. For a given identity, a derived clock skew is compared to its corresponding reference clock skew. A clock skew fingerprint is a constant which makes the similarity measurement fairly trivial. The clock skew similarity metric is equal to the difference between the derived clock skew s_v and the reference clock skew for the identity in question. A set error margin threshold will dictate whether or not the device will be authenticated and is in the magnitude of parts per million (ppm). However, there is no definitive consensus regarding the value of said threshold in order to uniquely identify a device [23] [26] [28].

3.2.4 Experiment Results

The author of [25] describes an experiment setup which was executed in an isolated lab environment consisting of four computers of same specification, communicating over wired connection. Under such conditions, it was concluded that an error margin of 1ppm had to be set to uniquely identify a device. However, the conditions used in the experiment does not reflect real-world conditions in public places with real wireless user traffic. The experiment in [28] was performed in numerous locations, all of them public places with a set of participating devices in the magnitude of hundreds. A sample size of 300 packets recorded under 30 seconds was

needed to generate a clock skew with 0.5 ppm precision. The experiment showed that 88% of all tested clock skews managed to reach at least 0.5ppm error margin.

3.3   Inter Arrival Time

Packet inter-arrival time (IAT) analyses the inter-arrival time in a sequence of packets in order to derive a device fingerprint. Since IAT is based on the time of reception, the technique is not dependent on timestamps found within the packets themselves. The implications of this is that IAT analysis is transport layer agnostic [30] [31]. Research seen in [31] pioneered packet IAT fingerprinting. The research showed how the technique could classify the device type of a proving AP i.e. the manufacturer and model of the AP. However, the technique was not able to uniquely identify a device. Hence, the research was extended in [30] where the device type identification is complemented with machine learning techniques to arrive at a unique device identity.

3.3.1   Data Collection

In [31] [30], IAT fingerprinting is done against wireless network APs to detect rogue APs. Data is extracted from the AP by generating a packet train at one of the wireless network nodes. The packet train emulates normal but intense network traffic and is sent to put the AP in question under stress. The corresponding output from the AP is recorded, especially the packets' shift in time. It is the shift in time between the output's packets that constitutes the packet inter-arrival time.

3.3.2   Fingerprint Generation

Fast fourier transform is applied to the output of the data collection phase. The inter-arrival times of the corresponding output to the packet train constitutes an input signal which is used to generate a fingerprint to compare against known device type fingerprints [31] [30].

The fingerprint is generated by applying wavelet transform to the input signal. More specifically, the input signal is run through a high-pass filter and the corresponding output is saved as a wavelet transform coefficient and then used as input in a new wavelet transform. This is done in a recursive fashion until at least five coefficients have been produced. The generated coefficients constitutes the device type fingerprint. Through empirical studies it is shown that coefficients 2 through 5 are the values that show most variations between device types and are thus the ones from which one can most easily detect a deviating device fingerprint. In [32] [30], the authors extend upon this method by applying it to device identity as well. Thus, a system is derived which classifies device type and establishes device identity.

3.3.3   Similarity Measurement

The IAT analysis implementations [30] [32] use an approach where device type classification is used as a pre-procedure to establishing the actual device ID. This is done in order to reject illegitimate nodes at an early stage without having to execute the full authentication procedure. Thus, two authentication steps are executed.

1. **Device Type Authentication**. The device type of an authenticating node is established. Packet IATs are collected, a device type fingerprint is generated and matched against the reference database. The derived fingerprint must match a reference fingerprint above a set similarity threshold and will either be accepted or rejected based on this.

2. **Device Identity Authentication**. Based on the device type derived in the previous step, the authenticating node will be matched against reference device identity fingerprints of same device type. Similar to device type authentication, a given similarity threshold must be passed in order for the authenticating node to be considered matched against a reference fingerprint.

As stated, a similarity measurement is executed to match the fingerprint against a set of known fingerprints found in the reference database. It is done by cross-correlating derived coefficient d_i against known coefficient k_i where index $2 \leq i \leq 5$. E.g. each derived coefficient is compared to its corresponding coefficient in a known fingerprint. The output from the similarity measurement is a metric between 0 and 1 for each coefficient. In the case of device type authentication, if the derived fingerprint belongs to a specific device type, the similarity measurement for each coefficient will differ by no more than 10 parts per thousand. A similarity measurement between two differing device types will yield a deviation in the magnitude of 100 parts per thousand. In the case of device identity authentication, the similarity measurement must yield an output across all coefficients that is above a given threshold. In [32], empirical studies showed that a 90% similarity threshold would yield minimal decision error rate.

3.3.4   Experiment Results

In [30] two experiments were performed, one in an isolated test environment and one in the live network of a school campus during peak hours. For both of the experiments, packet trains were repeatedly sent to the proving devices. The packet train sent was crafted in such a way that it would yield a response from the proving party with the size of 2500 packets transmitted under a period of 0.4375s. In the former experiment, an average true positive rate of 80% was produced. In the latter experiment, the true positive rate decreased to 58%.

3.4   Received Signal Strength Indication

The Received Signal Strength Indication (RSSI) over a wireless network can be used as a means of fingerprinting. It is a technique which can be executed with

most standard wireless systems of today requiring no additional hardware. Furthermore, RSSI data can be read out from most standard wireless device drivers, further lowering the integrational effort of an RSSI based scheme.

3.4.1 Data Collection

The data collection in an RSSI scheme consists of monitoring the received RSSI level for each device on a packet basis [33]. I.e. for each received packet, the packet's corresponding RSSI level is recorded and the packet is annotated with the sending device. In its simplest form the packets are recorded by a single receiver. In such a case the data holds one dimension and the verifying party can only estimate the distance to the proving party. However, many schemes utilize several receivers in order to achieve multi-dimensional and fine-grained localization estimates [33] [34] [35] [36].

3.4.2 Fingerprint Generation

When building a device fingerprint based on RSSI techniques, one cannot derive a fingerprint based on a specific RSSI value. Noise, multi-path effects and environmental variance will affect the wireless channel in such a way that RSSI for a given device will yield significant variations between measurements. Instead a multitude of techniques have been researched to accommodate to wireless channel variance factors. The approach described in [34] [36] utilizes the Euclidean distance in signal space between clusters of RSSI values to differentiate devices. Signal space denoting the n-dimensional geometric space in which the recorded RSSI values exists. Another technique makes use of a vector of receiver RSSI values, the vector makes the device fingerprint in this case and is called a signalprint [33]. It is a simplistic approach to RSSI-based localization where the core of the technique is based on the claim that the RSSI of a static node will not differ by more than 5db between contiguous packets.

In [34], an approach is described utilizing k-means clustering of RSSI values. The technique is based on the distance in signal space between two devices. As described above, one or multiple receivers can be used to record RSSI for a given packet. In the case of a single receiver, the signal space is one dimensional. The recorded RSSI values for two devices whom are spatially separated, will yield two clusters with their respective centroids located at different values along the X-axis. This can be further extended to a two-dimensional signal space, where the clusters will be found in different locations on the plane. In this fashion one can add more receivers in order to further differentiate devices in an n-dimensional signal space.

Research seen in [37] [33] gives an alternative approach utilizing signalprints as device fingerprints. A signalprint is a vector of arbitrary length where each cell represents an RSSI value from a given receiver. It is generated by aggregating the RSSI values from different receivers for a given packet transmission. The number of receivers must be larger than one. The signalprint can be used in two ways, either with the raw absolute RSSI values or with differential values. The latter is

a modified signalprint where the smallest value is subtracted from all remaining values in the vector. This way, the impact of RSSI oscillation is reduced. The experiment makes use of six spatially distributed receivers. This due to a larger number of receivers yielding a signalprint of greater accuracy. The authors of [33] argue that battery discharge will have a significant impact on the transmission power of a device. For this reason signalprints are used, since they in their differential value mode become agnostic of reducing transmission power of a battery driven device.

The authors of [36] registered the RSSI values of 1500 different indoor locations on 86 APs. The resulting data collection was then used as a training set for three different machine learning algorithms, namely, Random Forest, k-NN and JRip to assess which algorithm creates the most reliable device classification.

3.4.3   Similarity Measurement

In the case of using RSSI clustering [34] in signal space to differentiate devices, the collection of RSSI values for a given device acts as a fingerprint. A similarity measurement is done by calculating the distance between the centroids of the derived fingerprint and the corresponding reference fingerprint. If the Euclidean distance between the two is below threshold τ, the device is authenticated. The calculation of distance between the two RSSI clusters in a one-dimensional signal space is given by

$$\triangle P = 10\gamma log(\frac{d_2}{d_1}) + \triangle S \tag{3.1}$$

where P denotes the transmitting power, d_1 and d_2 represents distance of the two nodes to the receiver, τ is the path loss exponent and S is is the shadow fading, i.e. a PDF for attenuation due to radio channel distances. The equation for RSSI distance in n-dimensional signal space is as follows

$$\triangle D = \sqrt{\triangle P_1^2 + \cdots + \triangle P_n^2} \ . \tag{3.2}$$

The challenge of determining authentication threshold τ is to find a threshold where the number of decision errors is minimized. This is done by comparing the compound PDF for two devices d_1, d_2 placed at the same location and another PDF where d_2 is relocated to a different location whilst d_1 remains stationary. The two PDFs will have their respective means in different locations along the X-axis but a certain overlap can be seen between the tails of the two PDFs. The overlap is used to produce a threshold where false negatives as well as false positives are minimized. Hence, if communication is initiated with a node whose claimed identity is a but produces an RSSI cluster whose distance to the reference cluster of a is above the set threshold, it will be considered illegitimate and subsequently rejected from the network.

The technique showcased in [37] has a twofold approach to detecting identity-based attacks. Firstly, a *max matching rule* is used to assert that two subsequent signalprints S_1 and S_2 are originating from the same device. Secondly, the max

matching rule is combined with a *min matching rule*. The latter determines if S_1 and S_2 have been sent from two different devices. A *max matching rule* is formulated as follows, signalprints S_1 and S_2 are assumed to originate from the same device if the number of max matches between them is greater or equal to N_{max}. A max matching occurs when two corresponding values found in S_1 and S_2 respectively, show a difference that is less than or equal to threshold τ_{max}. The rule notation for max matching is written as

$$maxMatches(S_1, S_2, \tau_{max}) \geq N_{max} .$$ (3.3)

The *min matching rule* is defined as, signalprints S_1 and S_2 are sent from the two different devices if the number of min matches between them is greater or equal to N_{min}. A min matching is seen when two corresponding values in S_1 and S_2 are greater or equal to threshold τ_{min}. The min matching rule is based on the fact that even though RSSI values from the same fluctuates, they usually do not see fluctuation above 5 dB. Therefore, if a larger difference can be seen at many receivers for two subsequent signalprints, the packets in question can be assumed to arrive from two different devices. Similar to max matching, the rule notation for min matching is

$$minMatches(S_1, S_2, \tau_{min}) \leq N_{min} .$$ (3.4)

In [36], authentication was achieved on a packet basis. Meaning, every incoming packet was classified using the three different machine learning models. As only classification methods were used, every incoming packet was mapped to one of the 1500 locations registered during the fingerprint generation. Because of this, a packet sent from a previously unseen location would still be classified as belonging to one of the registered locations.

3.4.4 Experiment Results

In [34], the experimental setup consists of five receivers evenly distributed across an office floor where nodes located in 94 different locations have been placed. In order to evaluate an impersonation attack, a pair of nodes were randomly selected, where one acts as the compromised node and the other one as the impersonating node. Since the pair of nodes were chosen at random, the distance between them would vary between executions. However, the summarized result shows that for a false-positive rate less than 5%, a 97% detection rate was achieved. When performing tests with a 6.1 meter distance between the two nodes, the detection rate reduced to 90%.

The experiment performed in [37] had a similar experiment setup to the one seen in [34]. Distributed over an office floor, were 135 nodes acting as proving parties. Six receivers were used to capture RSSI data. The experiment did not explicitly measure detection of a spoofing device. Hence, no detection rate metric was produced. Instead, the experiment evaluated the number of false positives produced given a set of matching rules. For matching rules where $maxMatches = (S_1, S_2, 5) \geq 5$ and $minMatches(S_1, S_2, 10) = 0$ a false positive

rate of 1.1% was produced. The fraction of false positives mapped to the distance between S_1 and S_2 when it was observed can be seen in Table 3.1.

Table 3.1: The fraction of false positives mapped to distance between nodes when $maxMatches = (S_1, S_2, 5) \geq 5$ and $minMatches(S_1, S_2, 10) = 0$ is applied.

Pair distance	False positive occurrences
$\leq 10m$	99.0%
$\leq 7m$	90.7%
$\leq 5m$	72.2%

Results in [36] showed that Random Forest was the highest performing out of the three algorithms tested. It achieved a classification accuracy, or false negative rate, of above 90%.

3.5 Physically Unclonable Function

A physically unclonable function (PUF) maps a set of challenges to a set of responses based on a complex physical system [38]. PUFs are based on characteristics of the integrated circuit (IC), such as memory elements, logic delays, resistance and so on [39].

PUFs were first mentioned in [40], where it is proposed that manufacturing process alone create enough variations in the IC characterizations to be able to uniquely identify an IC. As a result of this, a PUF utilizes this uniqueness to provide randomness, i.e. given specific input, the PUF will generate unique output. Today a lot of different versions of PUFs are being used and researched, e.g. Arbiter PUF, Ring Oscillator PUF, SRAM PUF, Butterfly PUF, Flip-flop PUF, Bistable Ring PUF and more [41].

PUFs can, as described in [42], be divided into two primary types - strong PUFs and weak PUFs. Strong PUFs are capable to handle a large number of challenge-response pairs, in contrast to weak PUFs, which results in them being suited for authentication purposes. Weak PUFs on are the other hand better suited for key storage and key generation.

3.5.1 Data collection

The data collection, or enrollment phase, used by PUF is rather an active mapping of challenge-response pairs than a data collection. The verifying part generates random data, that is sent to the proving device as a challenge. The challenge is then used to generate unpredictable output, a response, via use of the PUF that is sent back to the verifying device and mapped to the sent challenge [43].

3.5.2   Fingerprint Generation

The challenge-response pair generated from the enrollment phase are stored in the verifying part [43]. Thus, the PUF fingerprint consists of a number of challenge-response pairs, where the challenges are randomly chosen by the verifying part and the responses are generated by the PUF in the proving device.

3.5.3   Similarity Measurement

During authentication, the verifying part sends one of the previously stored challenges to the proving part. If the response is not equivalent to that of the stored challenge-response pair, the proving part is not authenticated [43]. To prevent false positives, a number of challenges should be sent to the proving part [40].

3.5.4   Experiment Results

In [38] experiments which allowed a maximum 10 out of 128 bits to be incorrect to authenticate a device showed promising results. A false positive rate of $2.1 * 10^{-21}$ and a false negative rate of $5 * 10^{-11}$ was achieved.

3.6   Preamble Manipulation

Preamble manipulation is not a fingerprinting method in the sense that it can uniquely distinguish any kind of device. However, preamble manipulation can be used to separate different transceivers from each other [44]. Preamble manipulation utilizes that different transceivers, especially between manufacturers, respond differently to alterations in the physical layer preamble. The preamble is normally used to synchronize transmission timing between devices. It consists of a known set of bits that indicate start of frame.

3.6.1   Data Collection

In [44] and [45] a USRP Software Defined Radio (SDR) is used to capture packets and modify the preamble. Different alterations of the preamble are then produced and sent to a collection of different transceivers. A number of packets with compulsory replies, such as beacon requests, are then sent to the proving transceivers. For every packet with a manipulated preamble sent there are two possible outcomes, either a response is sent back or the packet is ignored. Thus, for every preamble combination the proving transceivers can be divided into two groups, one with those that, despite the manipulation, responds to the packet and one group with those that do not.

3.6.2   Fingerprint Generation

Many preamble alterations are then tested and based on the groups generated from the data collection a decision tree can be created. The fingerprint of any given transceiver can hence be described as a node in the decision tree.

3.6.3   Similarity Measurement

Devices can then be identified through the previously created scheme. This is done by starting in the root of the decision tree and then following the path created to the target node. If the device responds to each manipulated preamble in accordance to the decision tree, the device is authenticated.

3.6.4   Experiment Results

As preamble manipulation is based on a decision tree rather than matching a generated fingerprint with a reference fingerprint the results are not comparable with other fingerprinting techniques. Furthermore, the results presented in [44] and [45] are preliminary and more extensive testing has to be done to properly evaluate the method.

3.7   Radio Frequency Fingerprinting

When two devices communicate on a wireless channel they send data by transmitting radio signals. When said radio signals are being generated several hardware components are being used, for example DA converters, power amplifiers and radio frequency oscillators. Any variety in said components will slightly alter the generated radio signal [39]. This makes radio frequency fingerprinting (RFF) independent of the location of the transmitting device, as long as it is in communication range of the receiver. However, the ability to identify devices based on RFF is strongly affected by the SNR on the channel. As even little noise can affect the signal characteristics used for authentication [46], thus, longer distances on noisy channels can increase the error rate.

There exists two different categories of RFF, transient based and modulation based [47]. Transient based RFF utilizes differences in the turn-on/off transient of a radio signal. Modulation based RFF does instead utilize differences in the part of the signal that constitutes the transmitted data. Differences in the data part of the signal include frequency, magnitude and phase errors due to differences in the physical layer for example.

3.7.1   Data Collection

The differences in the transmitted signal caused by the hardware components can be used to identify devices in a wireless sensor network [48] [49]. Attributes that can be extracted from the RF signal, and used for identification, include [50]:

- transient waveform

- instantaneous phase and amplitude of the signal

- the signal tendency

- transient ramp profile in a temporal range

- the evolution in time

To be able to extract such attributes, most researchers have used software defined radio platforms such as USRP N210 [24].

3.7.2   Fingerprint Generation

To avoid having to store a lot of data on the verifying part and to fingerprint a device by individual RF characteristics a feature extraction can be done where redundant information is removed. One feature selection technique that can achieve this is Empirical Mode Decompositions first introduced in [51] and described in [46]. In Empirical Mode Decompositions the signal is decomposed into a set of Intrinsic Mode Functions. Intrinsic Mode Functions are a finite number of sub-signals with different frequency band, time and frequency distribution that together represent the whole signal. Thus, the differences in the Intrinsic Mode Functions can be viewed as differences in the signal. Other RFF extraction methods include Fourier transform and Wavelet transform [49].

3.7.3   Similarity Measurement

Authentication is done by matching a testing fingerprint with a reference fingerprint previously stored in the verifying part. There are a number of ways to achieve this, in [46] Transient Frequency Distribution is used to compare different Intrinsic Mode Functions. In [49] methods such as Norm Difference and Linear Discriminant Analyses strategy are described.

3.7.4   Experiment Results

Experiments in [24] showed that it was possible to identify a transceiver through RFF given the SNR at the receiver is 15dB or higher, the experiments used seven transceivers of the same model. In [52] RFF combined with Bayesian filter achieved an average true negative rate of 94-100% intrusion detection, however, they do not mention the false negative rate or differences caused by SNR. The results in [53] presented a true negative rate of above 90% at an SNR of 12 or greater at the receiver.

3.8   Radio Frequency Distinct Native Attribute Fingerprinting

All devices leak electromagnetic energy during normal operation that can be captured by – for example – a near field probe connected to an oscilloscope [54]. Radio Frequency Distinct Native Attribute (RF-DNA) fingerprinting is based on the fact that such electromagnetic emissions can be used to uniquely identify a device. The emissions can be both intentional and unintentional, where unintended emissions are generated by components on the IC [55].

3.8.1   Data Collection

The RF-DNA fingerprint consists of attributes of the RF emissions such as amplitude, phase and frequency. In [56] and [57] these attributes, or characteristics, are

derived individually and the RF emissions are transformed into the attribute domain in question. The statistical features – standard deviation, variance, skewness and kurtosis – are then generated.

3.8.2   Fingerprint Generation

The features generated in the data collection are combined to form one sample, multiple samples are then collected to generate a device fingerprint through statistical methods such as MDA training, which is described in [57].

3.8.3   Similarity Measurement

Devices can then be classified, or authenticated, as belonging to a specific group of devices or in a device verification situation where a rogue device impersonates a trusted device.

3.8.4   Experiment Results

Experiments [56] [57] showed that it was possible to correctly classify a device 90 percent of the time with an SNR of 12dB in inter-manufacturer cases and with an SNR of 16dB in intra-manufacturer cases.

Attacks Against Fingerprinting Techniques

In this chapter, attacks designed to thwart the fingerprinting techniques listed in Chapter 3 will be discussed. Attacks against fingerprinting techniques aim to replicate the behavior of a legitimate device in order to forge its corresponding fingerprint [32].

4.1 Clock Skew Replication Attack

Malicious entity Eve can observe network traffic between Alice and Bob. Furthermore, she can collect timestamps and calculate the clock skew of Alice. Once the clock skew of Alice is captured, Eve can replicate the clock skew to cheat Bob [58]. The replication is done by first calculating r according to (4.1), where s_1 is the skew difference between Eve and Bob and s_2 is the skew difference between Eve and Alice.

$$r = \frac{1 + s_1}{1 + s_2} \tag{4.1}$$

Every altered timestamp (t'_i) is then calculated by (4.2), where t_1 is the first timestamp.

$$t'_i = t_1 + r(t_i - t_1) \tag{4.2}$$

There are more complex ways to calculate the altered timestamp that also can handle changes to the sending period. However, they are based on the same principle [58].

4.2 Packet IAT Attack

In [30] an approach to forge a packet IAT fingerprint is described. In the attack, an adversary has to compute the PDF of its own network communication along with the PDF of the legitimate node which is to be impersonated. Once the two PDFs have been obtained, the difference between them has to be derived. The PDF difference will dictate how the adversary will have to change the frequency of its packet transmissions in order to emulate the behavior of the legitimate device. To emulate network behavior such as packet rate, an emulation tool like *netem* can be used.

4.3   Attacking RSSI

RSSI fingerprinting uses spatial correlation to distinguish nodes [34]. I.e. nodes are identified based on their respective spatial location. Thus, the success rate of RSSI-based replication detection is dependent on how capable the RSSI fingerprinting system is when it comes to distinguishing devices that are in close proximity to each other [59]. Experiment results in [34] show that detection rate decreases as distance between adversary and the node being impersonated shrinks. When the distance between the two nodes is 4.6m, the detection rate can fall as low as 55%. Thus, an adversary could place an illegitimate node close enough to the node being impersonated in order to bypass detection. However, if multiple receivers are used at the verifying party to measure the RSSI of a proving party, a multi-dimensional RSSI fingerprint is achieved. To successfully impersonate a node through the aforementioned approach, rather than matching only one RSSI, the adversary is now required to match all RSSIs registered at each antenna respectively.

4.4   Attacking RFF

In RFF fingerprinting signal characteristics unique to a device are captured by analyzing a device's wireless communication at the physical layer. In [60] two RFF impersonation attacks are described, *Impersonation by Feature Replay* and *Impersonation by Signal Replay*.

4.4.1   Impersonation by Feature Replay

The approach is designed for modulation-based RFF fingerprinting and makes use of an USRP SDR to craft a signal where the features extracted from the signal of the legitimate device are replicated [60]. More specifically, the SDR is used to modify parameters frequency offset, SYNC correlation, I/Q origin offset, magnitude and phase errors of the frame. The modification of the parameters are done according to recorded communication data of the device to be impersonated. E.g. to produce the correct frequency offset, an analysis is done of the power spectrum density of the signal transmitted by the target device. One can derive the carrier frequency from such an analysis and from this set the frequency offset of the adversary node accordingly. The remaining parameters are set in the same fashion.

4.4.2   Impersonation by Signal Replay

The signal replay attack can be used for transient-based and modulation-based RFF fingerprinting respectively. In contrast to the feature replay attack where a signal is modified to match the signal transmitted by the target device, a recorded radio packet is replayed in full [60]. With the help of a high-end arbitrary waveform

generator [61], the recorded packet can be regenerated and subsequently retransmitted with high precision. Impersonation by signal replay needs not know the underlying signal features which are used to generate a fingerprint at the verifying party, it simply replays recorded packets in their entirety. Signal replay can be further extended. By replaying parts of a recorded packet, a crafted payload can be embedded in a packet which utilizes parts of a previously recorded packet. However, in such cases the adversary requires in-depth knowledge of the specific features relevant to the fingerprint generation in order to successfully impersonate a legitimate node with a crafted packet.

4.5   Attacks against preamble manipulation

As preamble manipulation only identifies different transceiver types, an attacker with knowledge of the target system can use a similar transceiver to authenticate [44]. Additionally, an attacker can through the use of an SDR read the preamble and selectively ignore manipulated preambles to imitate the behavior of a specific transceiver.

4.6   PUF impersonation attack

Despite its name, research have shown that PUFs are in fact cloneable [62]. A PUF can be reverse engineered by estimating the various delays at the different logical gates in the IC. From the gate delay estimations, it is possible to synthesize an IC which reproduces the same behavior as the target IC. However, measuring delay at gate-level requires high-end and fine-grained equipment which would result in a high cost of executing such measurements. Alternatively, a different approach can be seen in measuring the global power leakage consumption and modify factors –which has a direct effect on global power leakage consumption– such as the input vector of the IC, supplied voltage and thermal conditioning [63]. By doing so, multiple linearly independent equations can be produced in order to formulate a system of linear equations from which the delay of each gate can be derived [62] [64]. It is assumed that the attacker has obtained a detailed list of the PUF's circuit design. Thus, a circuit design copy in combination with gate-level delay estimates is used to emulate a PUF.

4.7   Hill-climbing attack

A hill-climbing attack is an attack that can be harmful to most fingerprinting techniques as well as biometric recognition systems [65]. The attack is based on a repetitive submission of data and can be compared to a more intelligent version of brute-force attack that requires less memory [66]. In contrast to a brute-force attack, a hill-climbing attack only needs to store the sent data that is closest to the authentication threshold [67]. The attack requires that the adversary has a attained a side-channel to the verifying party where a similarity score can be read for each authentication attempt. A similarity score telling the distance between

the most recent attempt and the authentication threshold. If the value was closer to the threshold than the previously saved, the new value is saved as a template. Following messages are then slightly altered versions of the template message [65].

4.8 Jamming and Denial of Service

Any fingerprinting technique that relies on features in the radio signal, such as RFF, RSSI and RF-DNA, are vulnerable to jamming as Eve can transmit a weak jamming signal to alter the characteristics of Alice's signal enough to prevent Alice from authenticating [48] [47]. If the jamming signal is stronger, it can disrupt the whole network and sabotage the authentication process no matter what technique is used [68].

4.9 Attacking the learning phase

If the learning stage is polluted by Eve, i.e. if packages sent from Eve is included in the fingerprint generation of Alice, the reference fingerprint will be sabotaged [32]. This could lead to Eve being able to authenticate as Alice even with the added security layer.

Evaluation of Fingerprinting Techniques

This chapter will discuss the various fingerprinting techniques with respect to feasibility to implement given the limited time frame, ability to distinguish different devices, equipment requirements and replication effort. The outcome of the evaluation will be a listing of the various fingerprinting techniques in order of maturity and applicability to perform fingerprinting in a static WSN.

5.1 Feasibility

The different fingerprinting techniques show various degrees of implementational effort. Hence, feasibility to implement a given fingerprinting technique is heavily dependent on the short time frame of the project. Below the implementational effort will be discussed. Given the software engineering background of the authors, solutions that require hardware alterations in the device, such as PUF, will be overlooked.

Clock skew has the advantage of being a software implementation. However, as the clock skew requires timestamps from the proving device as described in Section 3.2, TCP or ICMP must be used. As most computationally restrained systems do not implement any protocol with timestamps, such functionality has to be implemented. This differs from IAT as, even though it also bases the fingerprinting on packets sent from the prover, it does not require timestamps from the transmitting device. Furthermore, a potential issue in both clock skew and IAT is where timestamping of a received packet should occur, i.e. when does the implementation interpret a packet as being received. If a received packet is timestamped in the upper layers of the stack, internal processes in the receiver such as queues in the RX-buffer could lead to unwanted delays. On the contrary, timestamping a packet in the lower layers of the stack will many times be more time-consuming to implement.

Preamble manipulation requires generation of a decision tree before it can be used to classify devices. To be able to generate a decision tree, extensive experimentation with a large number of transceivers has to be done as described in Section 3.6. An undertaking which would most likely consume the majority of the implementation time of the project.

RFF, RF-DNA and preamble manipulation all require external hardware to be able to identify devices. This could be difficult to integrate with the rest of the experimental setup. Additionally, RFF and RF-DNA both involve complex mathematical theory to extract and interpret features from a wireless signal. Implementing such algorithms is a non-trivial task which itself could most likely constitute a number of MSc thesis projects.

Problems could arise when implementing RSSI functionality as reading and integrating the RSSI value with the implementation could be more difficult than expected. Often times devices will present RSSI data for a network as a whole. However, achieving RSSI data on a per packet basis could also be difficult if it would prove to be a performance sink for network interface hardware in cases where network traffic is lively and numerous packets are received simultaneously.

5.2   Distinguishability

Each fingerprinting technique described in Chapter 3 provide different accuracy in classifying devices. The ability to distinguish one device from another can be compared to the uniqueness of a generated fingerprint. Fingerprinting techniques such as preamble manipulation and to some extent IAT are only able to distinguish between different device types, i.e. they are not able to separate devices of the same manufacturer and model. RFF and RF-DNA is able to identify individual devices during optimal network conditions, however, in systems with low SNR the ability to classify inter-manufacturer devices decreases significantly. Clock skew, RSSI and PUF are claimed to provide the means to differentiate on a device-level.

5.3   Robustness

Robustness is a measure of change of the fingerprinting data over time and in extension a measure of stability of the initially produced fingerprint. Robustness has a direct impact on the security of a system since the implication of low robustness is that a fingerprint produced by a device at time t will be difficult to reproduce for the same device at time t'. Hence, in the case of a fingerprint with low robustness, the reference fingerprint will have to be continuously updated in order to avoid a high rate of decisions error.

5.3.1   Data Variance

Major differences in robustness can be seen between fingerprinting techniques. The fingerprinting techniques IAT and clock skew found in Sections 3.3 and 3.2 are highly dependent on timestamps over network communications. Therefore, network conditions with high variance will result in unstable data on which to build a fingerprint. In [23], the stability of clock skews were evaluated. A packet trace of 38 days for 69 devices was divided into data sets of 12 hours. For the 12 hour fingerprints, the largest difference in clock skew for a single device during

the 38 days was 6.04 ppm. The mean for difference between 12 hour clock skews arrived at 2.28 ppm. Recall that in [25] an optimistic threshold to uniquely identify a device was set to 1 ppm. The mean variability for the 38 day experiment surpasses this threshold with more than a 100%. The literature summarized in Section 3.3 did not contain an explicit evaluation of the stability of IAT fingerprints. However, fingerprint variance can be inferred from the various detection experiment results found in the same section. The best test result showed an 80% true positive rate, whilst worst case was the true positive rate decreased to 58%. In comparison, the test results from [28] presented a true positive rate of 88%.

The robustness of RSSI-based authentication schemes varies among the different implementations [34] [33] [35] [37]. The nature of RSSI is an unstable one where a multitude of factors all have an impact on the received signal. The techniques described in Section 3.4 are all designed with noise and signal attenuation factors in mind. In [34] the remedy against signal attenuation factors is double. Firstly, the described approach makes use of several receivers. Secondly, the authentication is based on the Euclidean distance in signal space between the derived fingerprint and its corresponding reference. The distance threshold dictating the authentication of a node is determined through a rigorous process where the rate of error decision is minimized. Thus, the impact of noise and signal attenuation factors is kept to a minimum. The RSSI experiment results seen in Section 3.4 are overwhelmingly positive. However, a major limitation can be seen in RSSI fingerprinting, the spatial location of a device is used as its identity. The implications of this is that two devices located less than 6.1 meters from each other will be difficult to differentiate.

The authors of [37] presents the use of signalprints as a more simplistic approach to RSSI fingerprinting. A major positive is that the technique provides per packet authentication. Hence, the verifying party does not need to capture x number of packets before generating a fingerprint, which is the case in [34]. Instead, given properly set matching rules, a single packet will suffice in detecting an illegitimate node. However, the scheme draws on the sheer number of receivers rather than a rigorous statistical foundation when generating its fingerprint. A signalprint where the number of receivers is below six will produce a high rate of decision errors. One could argue that the approach is not as robust regarding the signal attenuation factors which [34] mitigates efficiently.

RF-DNA does in many cases show stable and robust results, however, some variance can be noted. In [57] results showed that the general true negative rate of unauthorized devices is relatively high at 90% or greater. Nevertheless, for a certain device in the experiment the true negative rate of 15% was recorded, showcasing that significant variance exists.

In [53], experiments with 3 smart meters and 3 RZUSBsticks as authorized devices and 3 RZUSBsticks as spoofing devices. The reference fingerprint consisted of data from 300 signals per device. Results showed that most of the spoofing attempts achieved a true positive rate of above 90% while keeping false positive rate under 2%. However, two of the spoofing attacks achieved false positive rates of 54% and 32% respectively, given a true positive rate of above 90%.

Based on the previous work included in the thesis project there is no data variance in the case of PUF and preamble manipulation described in previous research. However, as preamble manipulation fingerprinting is a classifier based on a manually defined decision tree, any transceiver which is not part of the tree will yield an error. This case could arise in a scenario of a heterogeneous network where a large number of transceiver types are communicating. It would be practically infeasible to generate a decision tree for that large a number of transceiver classes.

5.3.2   Environmental Effects

The experiment results obtained in [28] which can be seen in Section 3.2 shows that clock skew could function in real-life conditions. However, part of the experiment investigated variation of clock skew for a given device during the course of a day. Its results showed variations of 1-2ppm between clock skew measurements where significant temperature differences could be seen. These were variations which would far surpass the 1ppm error margin threshold to uniquely identify a device. IAT, which in comparison to clock skew performed much worse in terms of true positive rate and data variance, does not suffer from variations due to temperature changes.

RF-DNA and RFF is highly sensitive to the noise on the channel. Results in [53] shows that at low SNRs the average ability to correctly identify a device is very low, e.g. 50% true positive rate at 3dB SNR. Similar results can also be seen in the case of RF-DNA [57]. Furthermore, as the bit error rate is generally increased with low SNRs, authentication with PUF could be less applicable in low SNRs as it is based on number of bit errors as described in Section 3.5.4.

5.3.3   Device Aging

The scheme proposed in [34] has failed to account for the impact of battery discharge. Battery discharge has a direct effect on the transmitted signal power and a reference fingerprint will be invalidated given enough time. In this respect, signalprints [37] outperforms the RSSI spatial correlation approach. By utilizing differential signalprints instead of the raw absolute value equivalent, the approach is unaffected by the impact of reduced battery energy on transmitted signal power.

5.4   Additional Hardware

Some of the techniques listed in Chapter 3 require additional hardware. Additional hardware is in this case described as hardware components that is not included in a modern off-the-shelf personal computer. In the case of PUF fingerprinting, the physical unclonable function is actualized through hardware. The PUF is an IC mounted on the proving device itself, as mentioned in Section 3.5.

RF-DNA, preamble manipulation and RFF requires additional equipment to be able to extract the features that form a fingerprint, as described in Section 5.1. In

contrast to PUF where the additional hardware is internal, the additional hardware in this case is achieved through external equipment. It comes in the form of external radio signal capturing devices, e.g. an USRP SDR platform. IAT, RSSI and clock skew can be implemented without any additional hardware or hardware alternations.

5.5   Attack Persistence

As described in Chapter 4 there exist attacks against most of the fingerprinting techniques presented in Chapter 3. Most of the attacks are based on replication of the legitimate fingerprint.

Replication attacks can be divided into two categories, those that require additional equipment and those that do not. The attacks listed against clock skew and IAT fall into the second category as they both require the attacker to alter the frequency with which they send packets – which can be done purely in software. RSSI replication does not require additional hardware either as it is sufficient to place an attacking node in proximity to the target node. RFF and preamble manipulation replication fall in the first category as they both require the adversary to be able to read features in the transmitted signals. In addition to that, RFF replication also requires hardware that can manipulate the transmitted signal to match that of the target node. PUF replication attacks, although theoretically possible as described in Section 4.6, can be seen the attack most difficult to pursue. It requires an adversary with in depth knowledge of the PUF-based authentication scheme, copies of the PUF circuit connectivity and high-cost equipment.

Additionally, there are disruptive attacks that can increase the probability of authentication errors from a legitimate device. Fingerprinting techniques that are based on features in the transmitted signal, such as RFF and RF-DNA, are particularly vulnerable to such attacks.

5.6   Evaluation Verdict

Based on the various aspects of this evaluation, it can be seen that the fingerprinting techniques have reached different levels of maturity in their respective research fields. Some are too unreliable to be included in a live authentication system whilst others could work in conjunction with other authentication mechanisms. The fingerprinting techniques will be enumerated by descending level of maturity.

1. **PUF**, the fingerprinting technique that outperforms the remaining techniques . This holds true across all features considered apart from feasibility. In regards to feasibility, the major drawback of PUF fingerprinting is that it requires an IC that has to be mounted during the manufacturing process as it is difficult to do so subsequent to deployment.

2. **RSSI**, it maps the location of a device to the device's identity. In a static network such an assumption is valid. The performance results of RSSI are

high, especially given many receivers working in parallel. However, draw-backs come in the form of difficulties in distinguishing devices which are closer than 6m from each other and its dependence on multiple receivers to record RSSI values per packet. This means that a large number of receivers will have to be distributed across the network so that several RSSI values can be recorded for each message sent from each node in the entire network.

3. **Clock Skew**, shows overall strong results in uniquely identifying a device. However, the technique lacks in robustness since environmental factors such as temperature has a large impact on the derived fingerprint during authentication. But if one could overcome such limitations, clock skew would definitely be a mature fingerprinting technique fit to be incorporated into a live security system.

4. **RFF**, research shows promising results but is dependent on a high SNR as well as short distances between proving and verifying parties. Hence, RFF-based authentication is infeasible in deployments where nodes are further than 2m from the AP. Furthermore, additional hardware in the form of a USRP or equivalent SDR is required at the receiving end. This proves a major financial obstacle if one were to incorporate it in an already deployed system since the cost of the additional hardware is in the order of thousands of dollars.

5. **IAT**, performance results will vary widely depending on device class. Certain device classes will contain devices which are easily distinguishable. Whilst other classes contain devices which prove to be indistinguishable. Furthermore, in an embedded scenario, IAT might be too power consuming due to its large number of packets under short time frame that need be collected in order to produce a fingerprint.

6. **RF-DNA**, looks very promising at first glance. However, when looking into details of experimental setups, it is clear that most of them are done in optimal environments, with short distances and powerful receivers. Thus, the applicability of RF-DNA is highly limited and does not fit with the WSN scenario discussed in the thesis project.

7. **Preamble manipulation**, is deemed as not suited for use in an authentication system in its current state. Not enough research with different transceiver types has been done to evaluate the technique properly. If many devices are to be considered when creating the decision tree, chances are that the tree would get too big and consequently too power consuming due to many round trips. Another disadvantage with preamble manipulation is the fact that it can not uniquely identify specific devices, only transceiver types.

The fingerprinting techniques to implement will differ from the top listings above. In the case of PUF it is not feasible to integrate it in to an pre-existing system, which is the topic of this thesis project. Furthermore, RFF seems promising to implement, however, as it requires additional hardware that can not be obtained given the limited resources of the thesis project. IAT seemed good at first glance,

however, it is designed for authenticating an intermediary node in a network such as an AP. The burst of packets IAT requires to provide reasonable results does not fit the constrained environment of the thesis project. RF-DNA and preamble manipulation both rate very low on the evaluation verdict hence they will not be tested. In conclusion the techniques to implement will be *clock skew* and *RSSI*.

Method

This chapter will chronicle the different phases of the experiment execution. Firstly identifying what requirements needs be fulfilled by the experiment system and how this informs the design and implementation of the testbed. Thereafter the trial and errors of the packet capturing process and lastly the implementation and testing of the different fingerprinting techniques will be described.

6.1 Developing the Testbed

Clock skew and RSSI-based fingerprinting techniques impose a specific set of requirements on the experiment system. Each respective fingerprinting technique operates on a specific set of attributes in the network traffic. Thus, the testbed must be designed in such a way that for each packet sent in the wireless network the different attributes can be easily extracted. The network traffic attributes in question are packet departure timestamp, packet receiver timestamp and packet RSSI.

Furthermore, in the case of clock skew, timestamp data of high resolution and accuracy is essential. Timestamp accuracy meaning that the timestamping occurs as close as possible to the physical event in question. For instance, one could argue that TCP packet timestamps do not satisfy the timestamp accuracy requirement. The time between injection of a TCP packet timestamp and the physical transmission of the packet in question can be highly variable [26]. TCP timestamps denotes the time when the operating system has scheduled the transmission of the packet. However, an arbitrary number of interrupts might occur between injection of timestamp and the actual transmission of the packet and thus the packet departure can be subject to a delay which is not reflected in the sending timestamp [26]. In contrast, wifi beacon timestamps broadcasted by access points are considered highly accurate, the timestamping occurs in direct conjunction with transmission of a packet when the network interface card has sensed the channel to be free. Similarly, the timestamp at the receiving end needs be generated close to the physical reception of the packet. From this, two more requirements are imposed on the system: accurate timestamping at sender and accurate timestamping at receiver. The requirements of the system are the following:

1. **Sending timestamp of packet**, used in conjunction with receiver timestamp to enable clock skew.

2. **Receiver timestamp of packet**, used by clock skew.

3. **RSSI of packet**, essential for generating signalprints and other RSSI based fingerprints.

4. **Accurate timestamping at sender.**

5. **Accurate timestamping at receiver.**

6.1.1 Technology Selection

The system requirements informs the selection of technologies which will comprise the experiment system. The selection of protocol stack was limited to ones found in the IoT domain. The main contenders being Thread protocol stack, Bluetooth Low Energy and regular Wifi.

The **Thread protocol stack** was a natural option, being used in the Tritech demo system with a tool suite readily available for use. However, when evaluating the API of the platform [5] on which Tritech's demo system is built, it was established that RSSI data on a per packet resolution could not be achieved without making extensive modifications to the internals of the platform. As with many regular desktop operating systems, the network interface hardware registers an RSSI upon receiving a packet. Nonetheless, the RSSI data is most often discarded before leaving kernel space [69]. As is the case with the demo platform. Moreover, the Thread protocol stack has yet to see widespread adoption in the IoT domain. Thus, it will not be compatible with a majority of off-the-shelf computers acting as a gateway devices. An external dongle is needed for a gateway to talk Thread. This has major implications on the accuracy of the receive timestamp. The external thread dongle does not include an arrival timestamp with the information forwarded to its host device, and therefore timestamping does not occur until the packet has traveled through the dongle and reached the kernel of the gateway device itself [70] which heavily decreases the accuracy of the timstamp.

In contrast, **Wifi** is a de-facto-standard in wireless home networks and does not face the same compatibility issues as Thread with off-the-shelf hardware, but similarly to Thread the wifi devices discard packet RSSI data before leaving kernel space. The wifi devices were all running a Raspbian distribution of Linux and the kernel could without too big an effort be modified to retain RSSI data and thus let it travel all the way to user space. However, by doing so one would put the devices in a monitor state where a device will listen to all packets sent in a network but it cannot send packets itself. For the means and purposes of the experiment, the devices were required to be able to both send and receive packets.

Although relatively new, the **Bluetooth Low Energy** stack has seen widespread use and is readily available as a standard communications protocol in many recent devices ranging from desktop computers to smartphones. The packet RSSI data

which is difficult to obtain in Thread and Wifi is a required header field in the
Bluetooth Low Energy standard. Thus, BLE transport satisfied the three network
attribute requirements send timestamp, receive timestamp and RSSI by default.
Furthermore, the BLE devices $nRF52832$ [71] available for use at the Tritech office
proved excellent in terms of accurate timestamping capabilities. The nRF52832
are specifically designed for BLE communication and they are development boards
running without an operating system. Therefore, the code does not run atop sev-
eral layers of operating system abstractions giving the developer more fine grained
control of when the sending timestamp will be generated and injected into the
packet.

Regardless of the transport protocol stack, the gateway of the experiment sys-
tem would be realized by a computer running the linux operating system. The
requirement of achieving an accurate receive timestamp would have to be fulfilled
by a linux available tool. Many of the research papers which this thesis is based
on [25] [32] utilize the pcap software to monitor their device network interfaces in
order to obtain receive timestamps of high accuracy. The pcap software leverages
the timestamps generated in the kernel of the receiving device. Furthermore, the
pcap software is able to listen to all of the three contending transport medias.
Therefore, accurate timestamping at receiver is satisfied.

Table 6.1: Transport media suitability

Protocol Stack	Send TS	receive TS	RSSI	Accurate send TS	Accurate receive TS
Thread	✓	✓	✗	✓	✗
Wifi	✓	✓	✗	✓	✓
BLE	✓	✓	✓	✓	✓

Table 6.1 shows the requirements met by each transport media. As can be seen,
BLE is the one satisfying all of requirements. For this reason Tritech's Thread
based demo platform was discarded in favor of a BLE driven system.

6.2   Testbed Overview

The system derived for the thesis experiment can be seen in Figure 6.1. It simu-
lates a wireless sensor network and consists of four sensors reporting sensory data.
The data is received by a number of spatially separated antennae who relay the
data to the gateway of the network. The gateway in turn forwards the data to a
remote machine for fingerprint generation. The computed fingerprints are stored
in the remote machine to be used for continuously authorizing all subsequent data
originating from the sensory nodes. In this way the remote machine acts as the
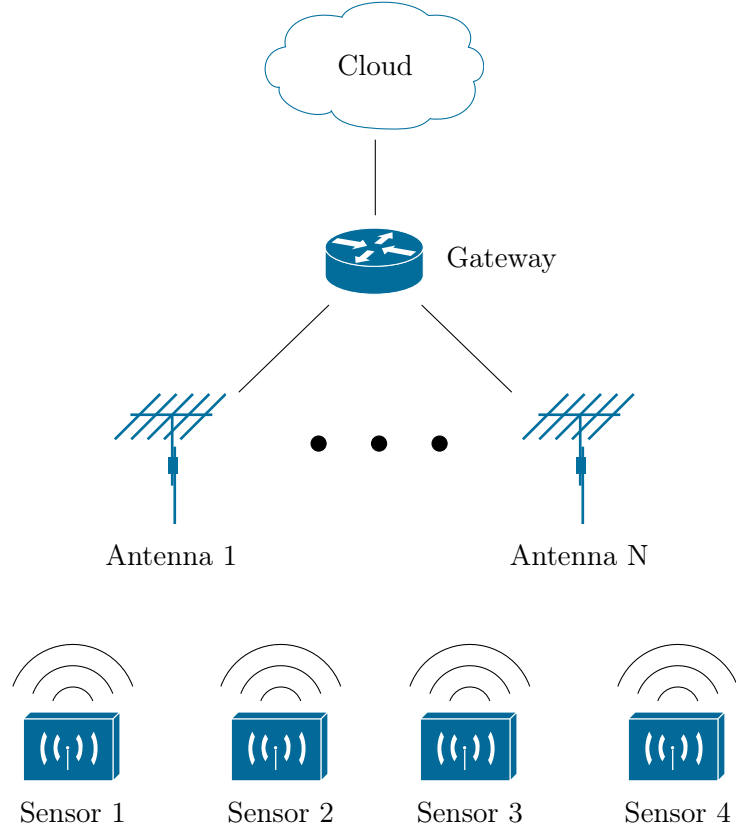verifying party and the sensory nodes makes the proving party.

Figure 6.1: Overview of experiment system

The experiment is conducted using four nRF52832 development boards to simulate the proving nodes of the network. The nodes are broadcasting BLE data at a constant rate consisting of their MAC-address as well as a sending timestamp. Raspberry Pi Zero Ws are used as relaying antennae to be able to register RSSI data at different positions. They read the RSSI of all incoming BLE packets and relay it to the main receiver through UDP, where the information is stored. Moreover, two additional devices are used to receive BLE data and produce receive timestamps of high accuracy. These devices acts as time reference in for the clock skew fingerprint. The two devices used for this are an inexpensive Raspberry Pi 3 and a more costly laptop equipped with an Intel i7 processor. The two are used to assess how the timestamping capabilities of a lower-end machine fares against a more high-end one. From this it can be concluded if a fingerprinting system can be realized with low-cost hardware.

A Raspberry Pi 3 Model B is used as gateway of the network. The gateway is the connection between the local network of nodes and the outside world. As previously stated, the computations —fingerprint generation and authentication—

are performed at a remote machine. In a real-life scenario this would be realized by a cloud service, but in the testbed a regular desktop computer is used for the computations.

6.3   Network Traffic Capture

The fingerprinting system is developed to meet the performance requirements of a live scenario. Still, network traffic capture data will be primarily used when implementing the fingerprinting techniques. To compare the different implementations of the fingerprinting techniques, one needs to achieve reproducible output from the system. Since the system input —network traffic— in this instance is highly variable, making use of network traffic captures is essential when evaluating and refining the different implementations. Moreover, one needs to be able to ensure that the authentication results of the experiments are dependable and not a product of temporary network conditions at the specific time of capture. For this reason, the collected data sets needs to be continuous captures, large in size, not only in the number of packets collected but also in their duration of time.

The testbed was setup according to Figure 6.2. The location was a home environment where the nodes in the testbed could operate without interference from other network devices. The length of the traffic captures varied between 12 to 24 hours.
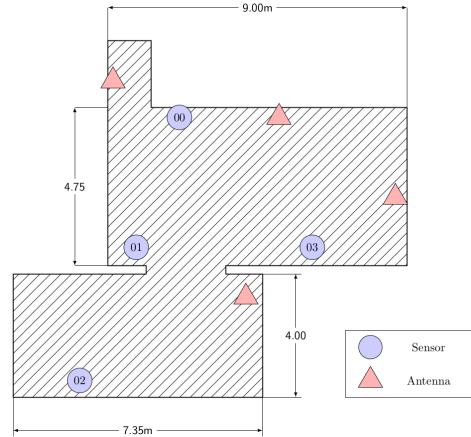

Figure 6.2: System layout

6.4   Fingerprint Implementation

The implementation and design of the different fingerprinting techniques followed the papers reviewed in Chapter 3. Remember that the generation of a reference

fingerprint is required for each device before an authentication can take place. Additionally, a certain data set size is needed in order to perform a reliable authentication of a device against the aforementioned reference fingerprint. The smaller the required data set size, the sooner it can be determined if a node is legitimate or not. Hence, the aim of the experiment is to minimize the number of packets required to generate a reference and authentication respectively whilst still producing reliable authentication results.

Once an initial implementation had been achieved, network traffic captures were used to evaluate the implementation. Evaluation of the different fingerprinting techniques would take the following steps:

1. Fingerprint specific parameters are set in order to trace what impact they have on the authentication correctness of the implementation.

2. A subset of the traffic capture is used to generate a reference fingerprint F_{ref} for each device in the network.

3. The remainder of the traffic capture is used to generate authentication fingerprints.

4. The true device ID of each authentication fingerprint F_{auth} is known, but in order to assess the reliability of the fingerprint implementation, F_{auth} is authenticated against all F_{ref}. Knowing the true ID of F_{auth} and F_{ref} one can immediately see if the fingerprinting implementation has decided correctly in its authentication verdict.

5. After a set number of authentications against the fingerprint implementation, the total false positive and false negative rate is calculated.

In this manner, the fingerprinting techniques were implemented and further refined. Between each invocation of a fingerprint evaluation, the fingerprint parameters would be modified. The false positive and false negative rate acts as a measure of the implementation's overall performance. To obtain a single metric on which to compare the different implementations, the f-score was derived from the false positive and false negative results.

6.4.1 Clock Skew

The clock skew implementation used was heavily influenced by [29] to compute the individual skews using offset between received and sent time for each packet. To detect impersonating nodes, two different methods were used, one threshold solution and one machine learning solution using Isolation Forest [21].

Threshold solution

Based on the research in Chapter 3, most previous experimentation with clock skew have used a threshold value of some kind to separate legit from corrupt nodes. Because of this, a threshold solution that takes the mean of a number of clock skews then checks if a packet to authenticate is within a certain error margin

was implemented. Different values of the error margin was tested to find the most suitable value.

Isolation Forest solution

In addition to the threshold solution a machine learning solution based on Isolation Forest was created. The reason for this was mainly the fact that it was difficult to visualize how a threshold solution for a combined fingerprint would be implemented. The Isolation Forest solution was because of this tested on the clock skew values alone as well to find the conditions under which it performs the best.

6.4.2   RSSI

Two RSSI-based implementations were derived based on the techniques described in Section 3.4, *signalprints* and *RSSI fingerprinting through unsupervised learning.*

The motivation behind the signalprint technique [37] is that node compromisation can be detected on a per packet basis with RSSI-based fingerprints. The thesis of the authors of [37] is that, even though highly variable, the RSSI of a static node will not differ by more than 5dB between subsequent packets. Thus, a variation larger than 5dB can be seen as a sign of node compromisation. A signalprint is a vector of RSSI readings from different antennae for a given network packet. As described in Section 3.4, rules were set for how strictly contiguous signalprints should be judged. When evaluating the signalprint implementation, the strictness of the signalprint rule was gradually changed from the most lenient of settings to the most rigorous, to find the setting that produced the highest f-score.

The second RSSI implementation was influenced by the findings in article [36] where a random forest classifier is used for classifying the location of moving objects in an indoor environment. In contrast to said system, the aim of the RSSI fingerprinting in the testbed is to assess the location of a static node. For this reason, a more simplistic implementation has been derived. The authors of [36] used single RSSI values as input to their positioning system to achieve a rapid system response. The system of the MSc project does not have as strict timing requirements. Thus, it can afford to take time to aggregate RSSI values for each respective antenna. It then computes the average of the aggregated values and uses it as system input. The aggregating and averaging of RSSI data is done in order to decrease the effect of variance in the RSSI data and consequently increasing the stability. To find the minimum viable data set size required to generate reliable RSSI fingerprints, the number of RSSI values aggregated was decreased between executions along with the total size of the training data set.

6.5   Combined Fingerprint

In the last step of the experiment, it was assessed if an authentication verdict produced by a combined fingerprint would perform better than the comprising

fingerprinting techniques individually. More specifically, the combined implementations were clock skew and RSSI fingerprinting through unsupervised learning. The two demonstrated best results during the individual evaluations. and were also the techniques which were most easy to reason about when considering a combined fingerprint.

In the combined fingerprint, Isolation Forest was used with clock skew as one feature and each antennae used as a feature each. Two experiments will be conducted, one where the clock skew and only the RSSI value originating from the gateway receiver will be used and one where the clock skew as well as the RSSI values from four receivers will be used.

Results

In this chapter, the results of from the experimentations will be presented. The chapter will be divided into three sections, where the results from the clock skew and signalprint experimentations will first be presented individually. Then the combined results will be presented.

The four nRF52 development kits used as nodes are identified by their MAC-address. The MAC-addresses share the same first seven bytes (c0:ca:fe:ba:be) and only differ in the last byte. Thus, the different nodes will be referred to as 00, 01, 02 and 03.

7.1 Clock Skew Results

As described in Section 3.2 the size of the data sample affects the accuracy of a generated clock skew. This section will present results from different aspects in the generation of clock skews as well as authentication methods based on clock skew.
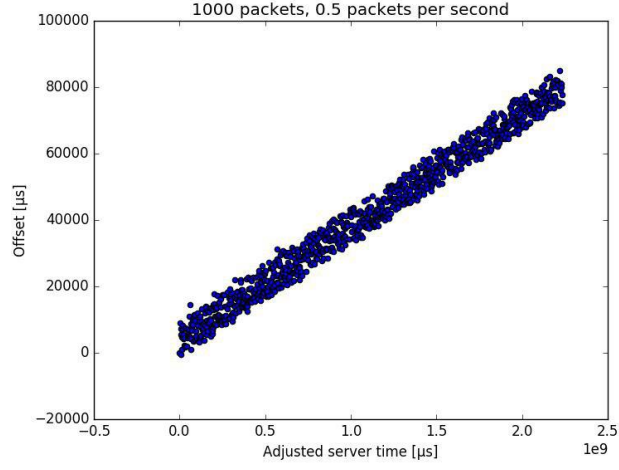
Figure 7.1: Data from 1000 packets sent from device 00

7.1.1   Data Gathering

As described in Section 3.2, the clock skew is calculated as the slope of the line generated by taking the linear regression between the offset and the server time for several packets. Figure 7.1 shows the data points gathered from 1000 packets with a packet interval of 0.5 packets per second.

7.1.2   Accuracy of Linear Regression

Table 7.1 shows the relationship between data sample size, packet sending frequency and the coefficient of determination (R squared) from the linear regression. To prevent effect of outliers, the R squared values presented in the table are an average of ten samples generated by data sent from device 00 rounded to four decimals.

Table 7.1: Relationship between linear regression accuracy, sample size and sending frequency.

| Number of packets per sample	Packets per second	Average R squared
100	0.5	0.3526
200	0.5	0.6735
500	0.5	0.9269
1000	0.5	0.9805
100	1	0.1042
200	1	0.3753
500	1	0.7805
1000	1	0.9360
100	2	0.0326
200	2	0.1287
500	2	0.4616
1000	2	0.7725

7.1.3   Clock Skew Generation

The clock skews of the four nodes are visualised in Figures 7.2a to 7.2b. The dotted lines represent the min and max clock skew recorded during the capture for each device. All clock skews presented in the figures are generated from the same packets, although with different sample sizes and sampling devices. The sending frequencies were all set to 1 packet per second.

7.1.4   Threshold Authentication Results

The results of the threshold authentication can be seen in Table 7.2. The ppm precision is calculated by taking the mean of N packets where N is the number of clock skews in the learning phase and then plus-minus with a variable D. A clock skew x will thus be authenticated if $N - D < x < N + D$. Note that the false negatives and false positives presented in Table 7.2 are average scores of all four nodes. Tests were done with varying N, however, no table change was observed. Thus, all values presented in Table 7.2 were conducted with $N = 10$.
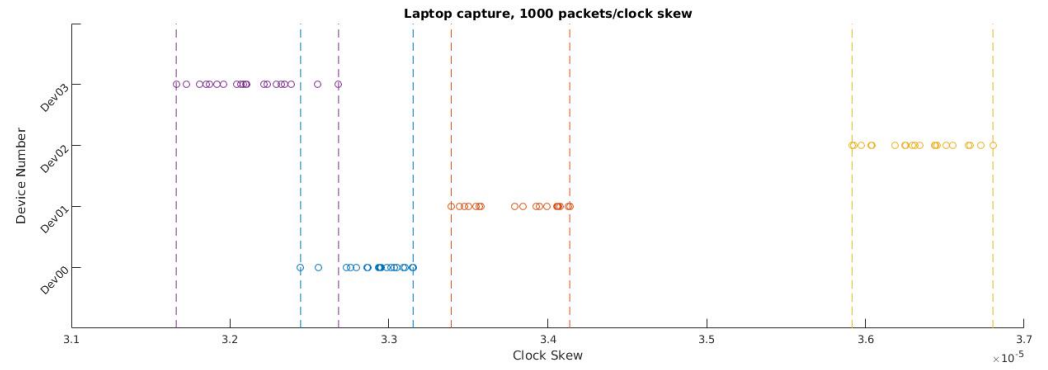
(a) Laptop capture with 1000 packets/clock skew. Tight clustering of clock skew values
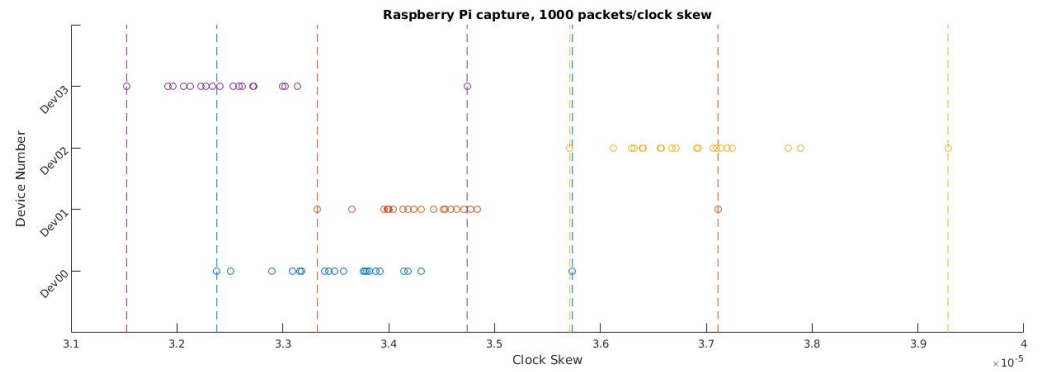
(b) Raspberry Pi capture with 500 packets per clock skew. Wide spread within clock skew clusters.
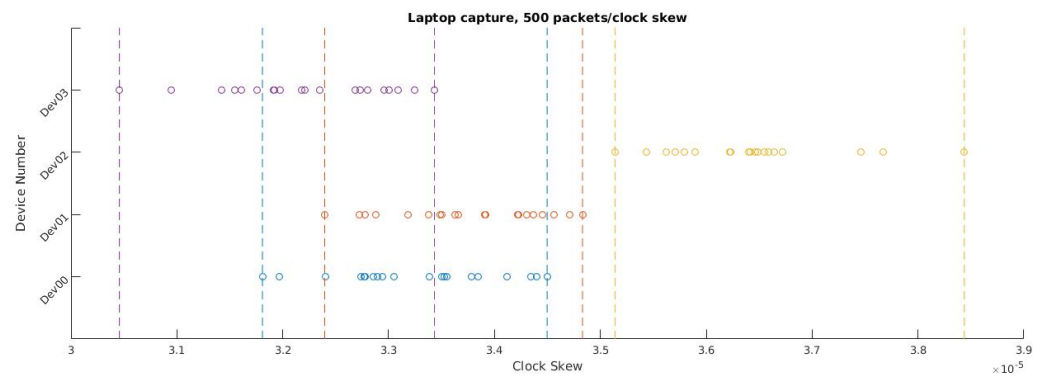
(c) Laptop capture with 500 packets per clock skew. Wide spread within clock skew clusters.

Figure 7.2: Clock Skew quality produced by different devices

Table 7.2: Clock skew threshold results

Packets per skew	D (1e-6)	False positives	False negatives
1000	0.25	0.005	0.435
1000	0.5	0.0175	0.145
1000	0.75	0.1125	0.0125
1000	1	0.1875	0.0
500	0.25	0.0425	0.78
500	0.5	0.0825	0.5625
500	0.75	0.13	0.3675
500	1	0.175	0.2125

Based on the results in 7.2 a test with $D = 0.8e - 6$ and 1000 packets per skew was conducted. The results of which are presented in Table 7.3.

Table 7.3: Clock skew threshold results per device

Device	False positives	False negatives	F-score
00	0.2	0.0	0.9091
01	0.2	0.0	0.9091
02	0.0	0.0	1
03	0.14	0.0	0.9346

7.1.5   Isolation Forest Authentication Results

An overview of the results from the isolation forest authentication experimentation is presented in Table 7.4. All false negative and false positive values presented are averages for all four devices.

Table 7.4: Clock skew results with isolation forest

Packets per skew	Skews per model	False positives	False negatives
500	20	0.4877	0.0833
500	40	0.2778	0.0967
500	60	0.3002	0.045
1000	10	0.7630	0.0611
1000	20	0.2725	0.0725
1000	30	0.0173	0.05

Table 7.5 shows the results of the isolation forest experimentation per device. The data is generated using 1000 packets per clock skew and 30 clock skews per model.

Table 7.5: Clock skew results with isolation forest per device

| Device	False positives	False negatives	F-score
00	0.0100	0.1143	0.9344
01	0.0421	0.0857	0.9347
02	0	0	1
03	0.0171	0	0.9915

7.2   RSSI Results

This section will present the results of the RSSI-based implementations. The results display the two implementations' performance in making correct authentication decisions given the setup presented in Figure 6.2.
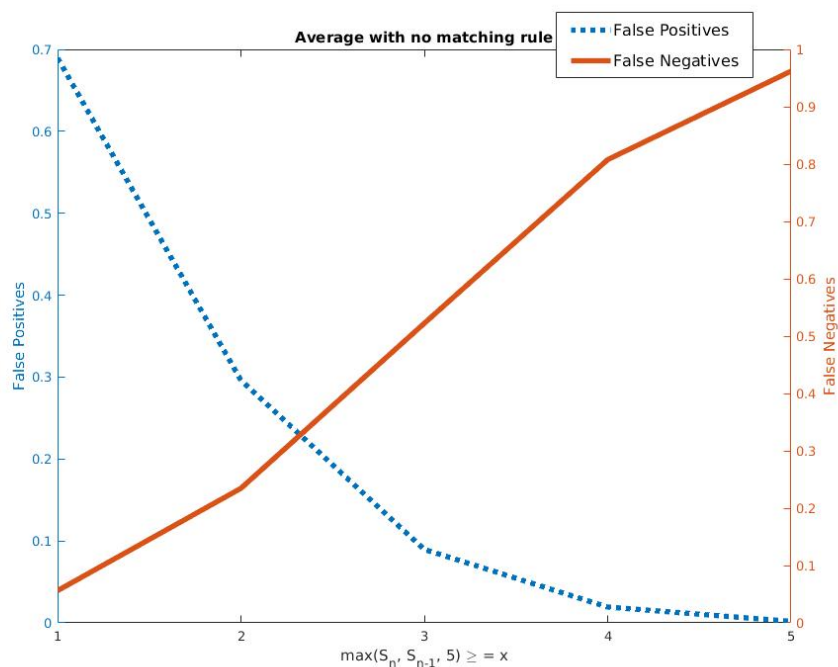
7.2.1   Signalprint

Recall that the signalprint technique applied a set of rules to help distinguish devices. The number of max matches for a pair of subsequent signalprints acts as a measure for the two signalprints originating from the same device. Conversely, the number of min matches provides an indication of how likely it is that a pair of signalprints originates from different devices. The rules applied will set a threshold for the number of max matches required and the highest number of min matches allowed.

The signalprint results obtained from the experiment data can be seen in Figures 7.3 and 7.4. Figure 7.3 shows how the system performs in terms of false negatives and false positives. Do note that the average for the system as a whole is presented rather than for each device individually. In 7.3a, no min matches are allowed and the max matches threshold is gradually increased and. In 7.3b, no restriction is applied to min matches and the max matches threshold is once again increased gradually. In 7.4 the f-score for each device is presented for the most strict and most permissive rule.
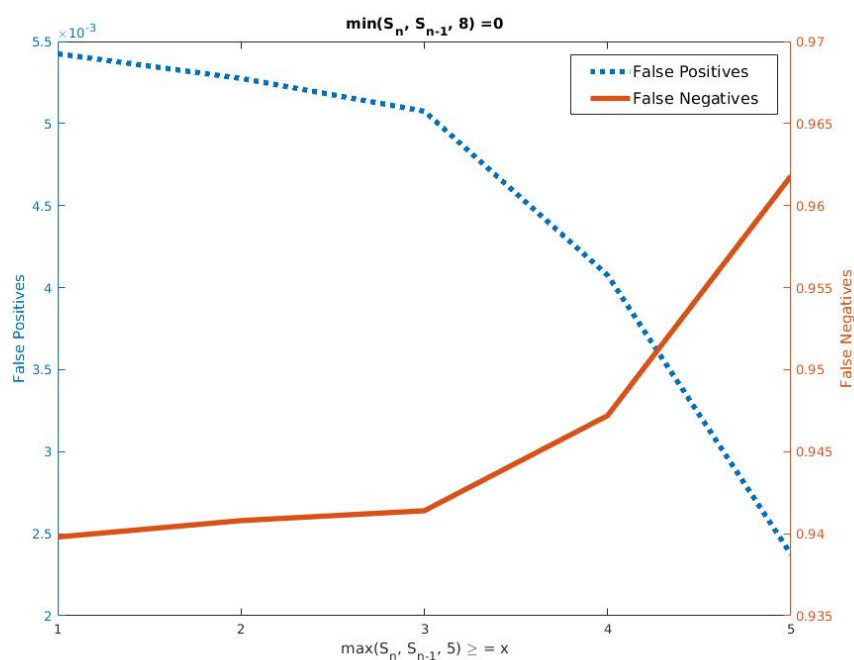
(a) Strict rules where no min matches are allowed.

(b) Permissive rules where min matches restriction is removed from the matching rule.

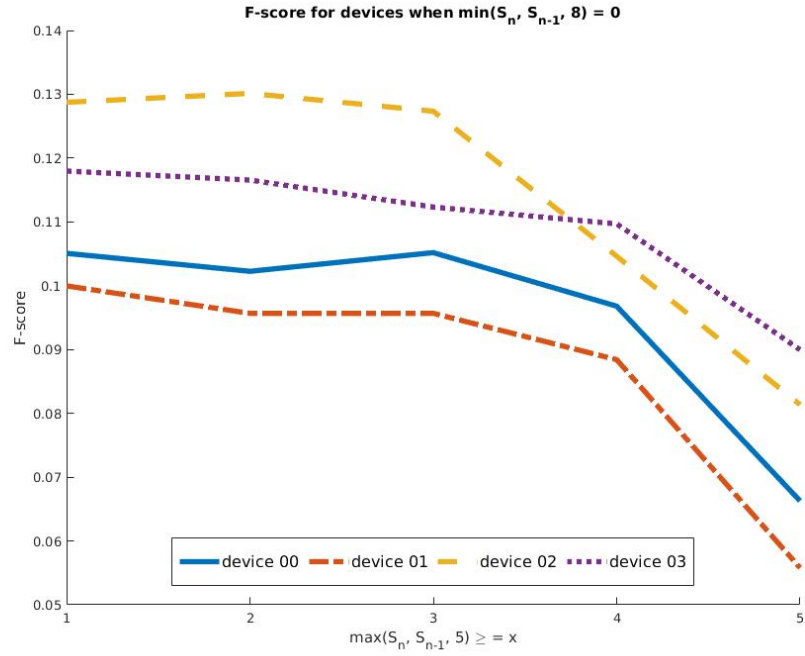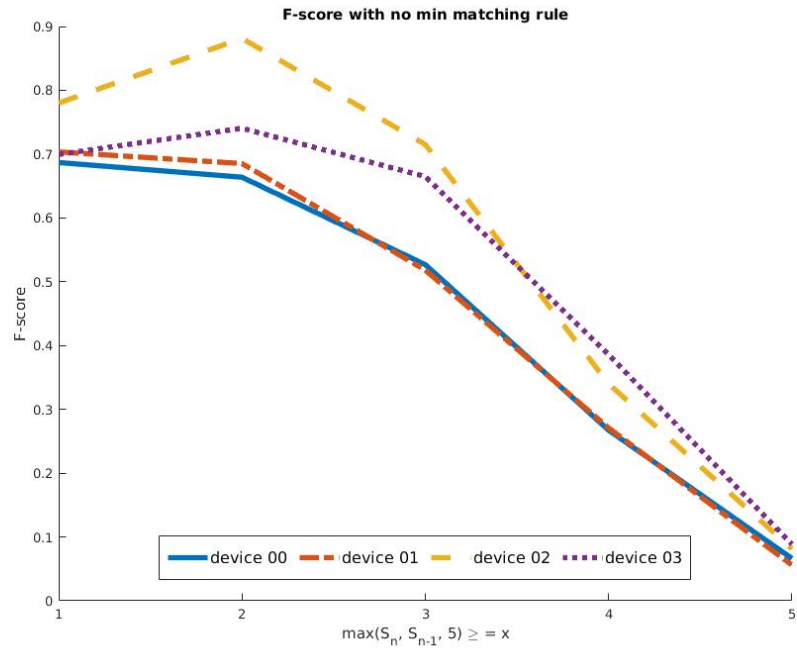**Figure 7.3:** Signalprint total average when applying different rules.

(a) Strict rules where no min matches are allowed.

(b) Permissive rules where min matches restriction is removed from the matching rule.

Figure 7.4: Signalprint f-Score when applying different rules.

In Table 7.6 the top results are displayed. For all devices, the system produced best results when no min matching rule was applied and the number of required max matches was equal to 2.

Table 7.6: The most performant signalprint results per device

Device	False positives	False negatives	F-score
00	0.3975	0.3058	0.6638
01	0.2757	0.3352	0.6852
02	0.1301	0.1096	0.8814
03	0.3812	0.1880	0.7405

7.2.2   RSSI fingerprinting through unsupervised learning

The RSSI through unsupervised learning implementation utilizes the mean value of a set of aggregated RSSI values. This is done in order to achieve more stable RSSI data to act as input for the unsupervised learning algorithm. The difference in fluctuation between the raw RSSI data and the corresponding mean data can be seen in Figure 7.5.
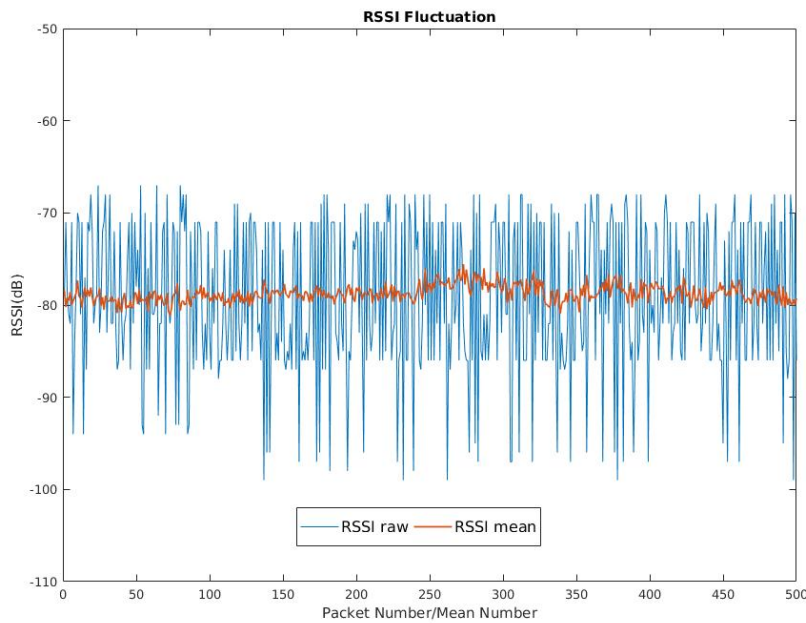

Figure 7.5: Fluctuation of raw RSSI and mean of 100 RSSI values.

The performance results for the RSSI implementation utilizing unsupervised learning are presented in Table 7.7. Do note that the false positive and false negative values are the derived mean for the system as a whole, not the individual devices. The *packets per average* column denotes the number of packets that were aggregated and averaged. The *averages per model* column represent the number of averages used to train the model.

Table 7.7: Isolation forest RSSI, system mean

Packets per average	Averages per model	False positives	False negatives
100	50	0.3640	0.0775
100	100	0.1353	0.1475
100	300	0	0.055
200	30	0.3309	0.1350
200	50	0.1975	0.0175
200	100	0	0.0225
500	20	0.0026	0.1925
500	30	0.0008	0.22
500	50	0.0009	0.3225
1000	10	0.1394	0.2643
1000	20	0	0.4143
1000	30	0.0450	0.3072

Table 7.8 shows the top performing results from Table 7.7 on a device basis. The mean value input to the unsupervised model was calculated from 200 RSSI values and in total 100 means were used to train the model.

Table 7.8: Isolation forest RSSI

Device	False positives	False negatives	F-score
00	0	0.01	0.995
01	0	0.04	0.9796
02	0	0	1
03	0	0.04	0.9796

7.3 Combined Results

The combined results can be presented in two different categories. First, the case when only the gateway is used to receive packets. The other case is when several antennas are used to register RSSI values from different positions. Only the best results from each experiment is presented. Both results are generated from 1000 packets per skew, 30 skews per model, 100 packets per average RSSI and 300

RSSI averages per model. As the feature vectors have different length, the clock skew feature vector is oversampled by duplicating entries. The results from the experimentation with one receiver is presented in Table 7.9 and the results with four receivers in Table 7.10.

Table 7.9: Combined fingerprint with only one receiver

Device	False positives	False negatives	F-score
00	0.01	0.0857	0.9503
01	0.0211	0	0.9896
02	0	0	1
03	0	0	1

Table 7.10: Combined fingerprint with four receivers

Device	False positives	False negatives	F-score
00	0	0	1
01	0	0	1
02	0	0.0571	0.9706
03	0	0	1

Discussion

In this chapter the results presented in Chapter 7 as well as future works will be discussed.

8.1 Test Bed and Data Capture

The test bed was created with a live scenario in mind. Because of this, a vast majority of the implementation effort has been spent on creating a system that captures live data, builds a model and then authenticates a device against said model. If instead a system that only worked with previously captured pcap files had been implemented, more effort and time would have been put into executing a more comprehensive experiment study of the fingerprinting techniques. As a more extensive study could have revealed factors with high impact on the implementations' performance, a more solid foundation of knowledge would have informed the process of refining the fingerprinting techniques.

Moreover, the system can accomodate to the addition of more nodes to the network. Functionality to move all calculations to a remote server has been implemented. Thus, The fingerprinting functionality is no longer constrained by the limited resources seen in the constrained devices of the local system. Additionally, the sensory nodes in the system perform few calculations as their sole task is to update their advertisement data with a timestamp. One possible bottleneck that could arise when scaling up the system is the gateway's ability to handle incoming packets. Furthermore, the memory requirements of the local system are relatively low as most data is stored remotely, in either cloud or as in the setup used during the experimentation, on a remote laptop.

However, the authentication decision making was originally planned to be performed at the gateway device whilst the model building would still be done at the remote server. By doing this, more swift feedback could be seen from the system as soon as an illegitimate device was discovered. Illegitimate data would be stopped at gateway-level and thus the remote server would never be reached by data originating from a corrupt node. However, it was deemed too taxing both computationally and memory-wise on the gateway device. Thus, all fingerprinting logic was moved to the remote server. It can be seen as a trade-off between sys-

tem security and resource consumption on the local system. The gateway is now
able to handle a larger number of nodes whilst the system feedback immediacy
is lowered. Furthermore, since data originating from an illegitimate node is no
longer discarded at gateway-level, there is now a possibility to perform the denial
of service attacks against the remote system.

Due to the limited time frame of this thesis project and the amount of time re-
quired to capture an adequate amount of data, there are a number of different
capture setups that could not be tested. For example, a capture in different types
of environments could be tested to understand how the system behaves in different
situations. Furthermore, all captures were done with completely static nodes, it
could be interesting to study how the system behaves if one or many nodes are
moved slightly during a data capture.

8.2   Clock Skew

As can be seen in Table 7.1, the accuracy of a generated clock skew is increasing
with the number of packets used in the linear regression. However, as there seems
to be no or very small differences if the number of packets are doubled but the
sending frequency of the packets are cut in half, it is reasonable to assume that the
accuracy of the clock skew is more reliant on the time between the first and the
last captured packets. Although, a very small number of recorded packets would
increase the effect of outliers.

The original test set up was aiming to use a Raspberry Pi model 3B as gate-
way. Early results showed that the accuracy of the clock skews, i.e. the ppm
differences between collected clock skews, were inadequate for device separation.
At first, it was thought that BLE – since it has, to the knowledge of the writers,
never been tried in a clock skew fingerprinting scheme – was the cause of the irreg-
ular results. However, when a laptop was used as a collecting device, the results
improved significantly. Thus, it is concluded that the hardware of the Raspberry
Pi is insufficient to be able to generate accurate clock skews. This is most likely
due to the inaccuracy of the clock used, but it could also be due to the network
capabilities of the Raspberry Pi for example.

As can be seen in Figure 7.1, the number of outliers in the data captures were
next to none. As most other papers on the subject experienced outliers this was
a bit surprising at first. Since the experiments done in this paper used BLE and
most other papers measured the clock skew via either WLAN or Ethernet this
is most likely the cause. One plausible explanation for this is the fact that the
number of BLE devices used are more uncommon than WLAN devices, thus, the
channel is less noisy.

The results shown in Table 7.2 show that authentication with threshold values
perform well. It is easy to see that a low value of D decreases the false positives
and increases the false negative and vice versa. The false positive result depends on

the clock skew of the other devices tested and thus difficult to generalize. Because of this, a system that generated virtually zero false negatives while still maintaining a low false positive rate was favorable. This resulted in Table 7.3 that, for the tested data sample, generated zero false negatives while still generating low false positive rates. Table 7.3 also shows that the false positive rate is affected by the results presented in Figure 7.2a. Device 02 has a clock skew that is separated by a higher ppm than the other devices and also generates the lowest false positive score.

Isolation forest results with clock skew as the only feature generally perform quite well as can be seen in Table 7.4. However, it does require a lot of data to build the model for stable results. Just as with the threshold experimentation, device 02 performs best as can be seen in Table 7.5, while devices 00 and 01 perform the worst in both cases. When comparing the two, the results presented show that isolation forest performs better given a large amount of data. However, if a small data sample is used to generate the model, the results suggest that a threshold solution is better.

The clock skew results are very much affected by the devices used. As different devices may or may not have different clock skews, an experiment with this small number of nodes could just as well have ended in the skews of all nodes being overlapped. Figure 7.2a shows that, while the differences between the devices are small (about 6ppm between the largest and smallest value), only devices 00 and 03 have overlapping clock skews given they are generated by 1000 packets. In another scenario, the devices could each be separated by >10ppm or all be overlapping. Because of this, experiments with a large number of nodes would have to be conducted to confirm the results. Another thing worth mentioning is the fact that the experiment only captured clock skews from devices of identical model and specifications. If devices from another manufacturer were to be used by a maleficent entity to attack the system, it is probable that the clock skew would be off by several ppm.

8.3   RSSI

This section will discuss the results obtained from the experiment study of the two RSSI-based implementations.

8.3.1   Signalprint

The experiment results showed that the signalprint technique was too unstable to be used in an authentication solution. At its most strict setting, when the system was configured to only accept network packets which it deemed likely to originate from a legitimate device, virtually all packets were rejected as seen in Figure 7.3a. For each device, signalprints delivers a verdict on each packet based on the immediately preceding accepted packet. This makes signalprints highly dependent on a continuous packet stream with minimal packet loss. However, when applying the most strict rule the largest factor contributing to the high rate of false negatives

is the no min matches requirement. The rule demands that no antenna may receive a packet where the perceived RSSI value differs by more than 8 dB from the preceding one, otherwise the packet is rejected. In Figure 7.5 it can be seen that this requirement does not work since the per-packet RSSI data sees fluctuations far surpassing 8dB.

As seen in Table 7.6, the best signalprint results were seen when applying no min matching rule and the required max matches were greater or equal to 2. This indicates that the max matches alone could be sufficient for distinguishing devices. Nonetheless, the measured decision correctness was still too low for signalprints to be used in an authentication system. This could have been further investigated in a testbed with a larger number of end devices and antennae. The signalprint research performed in [37] does recommend that the number of antennae be greater or equal to six. As the testbed was only equipped with five antennae, this could be a contributing factor to the poor results.

Signalprints operates on a per packet basis. Contrasting this with the clock skew and RSSI through unsupervised learning who operate on aggregate values, it was difficult to reason about how to incorporate signalprints in a combined fingerprint. Moreover, the poor experiment results of signalprints finalized the decision that the technique would not be further pursued.

8.3.2   RSSI fingerprinting through unsupervised learning

The aggregating and averaging of RSSI values before being input to an isolation forest model proved to be a highly effective way to identify and distinguish the end devices. By calculating the mean of a set of aggregated RSSI values, the violent oscillation in RSSI which can typically be seen in continuous RSSI readings was virtually cancelled out.

The results displayed in Table 7.7 shows how the number of aggregated packets along with the size of the model training set affects the decision correctness. The initial assumption was that the rate of decision correctness would grow as the number of aggregated values were increased. This proved to be an incorrect assumption. The experiments showed that the decision correctness would grow until the number of aggregated values was 200. From that point the the decision correctness saw a decline due to the rate of false negatives growing rapidly. When examining the data which gave rise to false negatives, it was seen that the authentication values differed from the extreme points of the training data by fractions of a dB. Remember that the raw data in this case is RSSI data which can oscillate well above 8dB. Thus, it is believed that the that the underlying isolation forest model was fed mean values which were clustered so tightly that subsequent authentication data would be rejected unless the values fell right inside of the training cluster.

8.4   Combined Fingerprint

The results presented in Tables 7.9 and 7.10 show that combining RSSI and clock skew yields a slightly better result compared to using the fingerprinting methods separately. However, a more large scale test would have to be done to definitely prove that our results hold. As the system requires 30000 packets to fit the Isolation Forest model, and the sending frequency used was only one packet per second, 24 hour data was only enough to do 35 authentication tests. Because of this, the data presented in the combined results constitute of 35 authentications. It is probable that the F-score will decrease slightly in a more substantial experiment.

Another advantage of using multiple fingerprinting techniques in a combination is the fact that the system as a whole will be more difficult to attack, seeing as an adversary would have to penetrate more layers of security.

One major disadvantage with using the fingerprinting techniques in a combined method is the amount of packets needed to separate legitimate and corrupt nodes. Seeing as a system that only utilizes RSSI would require less packets to detect anomalies.

The Sklearn API for Isolation Forest contains a lot of variables that can be chosen when generating a model. If a variable is not specified a default value is selected. Some experimentation has been done with especially the contamination parameter, which has been varied between 0 and 0.1. However, the results of this tweaking is not presented in the results as it would add an overwhelming amount of data. Because of this, only the best value for each individual result is selected.

As 1000 packets are used to generate one clock skew while only 100 packets are used to calculate an average RSSI, the feature vector lengths differ. To make them equal length, oversampling of the clock skew feature vector was performed. The oversampling was realized by duplicating the clock skew values. This could have a negative impact on the results as it could result in an over-fitted model. A more sophisticated oversampling method could be used to further improve the results.

8.5   Future Works

Due to time and financial constraints, many of the fingerprinting techniques listed in Chapter 3 were not tested. A more elaborate system containing additional fingerprinting methods could be created to achieve better and more robust results. Furthermore, more attributes could be taken from the existing fingerprinting techniques to be used as input to the unsupervised learning model. E.g. clock skew values are an aggregate of a set of packets, it is possible to derive statistical features such as variance and mean for the aggregate data set. Similar analysis could be applied to RSSI data to derive features beside the RSSI data itself.

As mentioned in the previous sections, the results seen in this thesis project needs to be confirmed in a more comprehensive study. Such a study would require a

larger number of nodes, preferably with a more heterogeneous set of devices to be fingerprinted. Additionally, an increased number of machine learning algorithms can be tested such as One-Class Support Vector Machine and Local Outlier Factor as well as a threshold based solution for RSSI averages.

To better get an understanding of how the system would operate in a live application, experimentation in a lively environment would be beneficial. For example in a busy factory with a large number of people in movement and where the position of the nodes may alter slightly due to bumps etc. Environmental factors should also be more closely studied, establishing what elements have a large impact on derived fingerprints will help in developing a more robust fingerprinting system.

Finally, experiments investigating attacks against the system created in this paper could be done. It is known, as can be seen in Chapter 4, that there exists numerous attacks against the fingerprinting techniques used in the system. Such a study would answer questions such as feasibility of fingerprinting attacks and give an indication of how much of an added security layer the system provides.

Chapter 9

Conclusion

This master thesis project has investigated how to authenticate a device in a wireless sensor network without the use of device identity credentials. The experiment study shows that device fingerprinting can be employed in order to deduce the identity of a device.

Previous research studied device fingerprinting techniques individually, in this thesis a novel approach has been taken where different fingerprinting methods have been combined to achieve better results. By utilizing the clock skew and RSSI of a device, it is possible to identify a device as legitimate or corrupt to a high degree of certainty. However, for accurate results, it has shown that a large data set is required to do so and instant discovery of corrupt nodes is not possible with the method used in the project.

Fingerprinting techniques are usually applied to environments where computational and memory resources are abundant. This project has shown how such techniques can be used in a more resource constrained environment. This is achieved by relocating resource demanding operations to a remote location, thus, lowering the resource demands on the devices found in the wireless static network. However, by doing so, the system takes a highly centralized character. Placing all computations and storage of data at a centralized server could however impose a security risk on the system.

Bibliography

[1] D Andeed. Smart grid security: Recent history demonstrates the dire need. 2013.

[2] Mark Ward. Smart meters can be hacked to cut power bills. `http://www.bbc.com/news/technology-29643276`. Accessed: 2017-03-01.

[3] Carl Hartung, James Balasalle, and Richard Han. Node compromise in sensor networks: The need for secure systems. *Department of Computer Science University of Colorado at Boulder*, 2005.

[4] Qiang Xu, Rong Zheng, Walid Saad, and Zhu Han. Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys & Tutorials*, 18(1):94–104, 2016.

[5] Efr32™ mighty gecko mesh networking wireless socs for zigbee® and thread. `http://www.silabs.com/products/wireless/mesh-networking/efr32mg-mighty-gecko-zigbee-thread-soc`. Accessed=2017-04-04.

[6] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.

[7] V Çağrı Güngör and Gerhard P Hancke. *Industrial wireless sensor networks: Applications, protocols, and standards.* Crc Press, 2013.

[8] Ieee standards association. `https://standards.ieee.org/findstds/standard/802.11ah-2016.html`. Accessed: 2017-05-04.

[9] Stefan Aust, R Venkatesha Prasad, and Ignas GMM Niemegeers. Ieee 802.11 ah: Advantages in standards and further challenges for sub 1 ghz wi-fi. In *Communications (ICC), 2012 IEEE International Conference on*, pages 6885–6889. IEEE, 2012.

[10] Johanna Nieminen, Carles Gomez, Markus Isomaki, Teemu Savolainen, Basavaraj Patil, Zach Shelby, Minjun Xi, and Joaquim Oller. Networking solutions for connecting bluetooth low energy enabled machines to the internet of things. *IEEE network*, 28(6):83–90, 2014.

[11] Clinton Francis Hughes. *Bluetooth Low Energy.* PhD thesis, ARIZONA STATE UNIVERSITY, 2015.

[12] IEEE 802.15 WPAN$^{\text{TM}}$ Task Group 4. Ieee 802.15.4 low-rate wireless networks, 2015.

[13] What is zigbee? `http://www.zigbee.org/what-is-zigbee/`. Accessed=2017-09-22.

[14] What is thread? `http://threadgroup.org/What-is-Thread/Connected-Home`. Accessed=2017-09-22.

[15] Martin Sauter. *From GSM to LTE: an introduction to mobile networks and mobile broadband*. John Wiley & Sons, 2010.

[16] Qutaiba Ibrahem Ali, Akram Abdulmaowjod, and Hussein Mahmood Mohammed. Simulation & performance study of wireless sensor network (wsn) using matlab. In *Energy, Power and Control (EPC-IQ), 2010 1st International Conference on*, pages 307–314. IEEE, 2010.

[17] Chris Guy. Wireless sensor networks. In *Sixth International Symposium on Instrumentation and Control Technology: Signal Analysis, Measurement Theory, Photo-Electronic technology, and Artificial Intelligence*, pages 63571I–63571I. International Society for Optics and Photonics, 2006.

[18] Sudipto Roy and Manisha J Nene. Prevention of node replication in wireless sensor network using received signal strength indicator, link quality indicator and packet sequence number. In *Green Engineering and Technologies (IC-GET), 2016 Online International Conference on*, pages 1–8. IEEE, 2016.

[19] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[20] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 413–422. IEEE, 2008.

[21] sklearn.ensamble.isolationforest. `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html`. Accessed=2017-08-18.

[22] Xiao Li, Ye-Yi Wang, and Alex Acero. Learning query intent from regularized click graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346. ACM, 2008.

[23] Tadayoshi Kohno, Andre Broido, and Kimberly C Claffy. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2(2):93–108, 2005.

[24] Saeed Ur Rehman, Kevin W Sowerby, and Colin Coghill. Analysis of impersonation attacks on systems using rf fingerprinting and low-end receivers. *Journal of Computer and System Sciences*, 80(3):591–601, 2014.

[25] Russ Fink. A statistical approach to remote physical device fingerprinting. In *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–7. IEEE, 2007.

[26] Suman Jana and Sneha K Kasera. On fast and accurate detection of unauthorized wireless access points using clock skews. *IEEE Transactions on Mobile Computing*, 9(3):449–462, 2010.

[27] Cherita L Corbett, Raheem A Beyah, and John A Copeland. Passive classification of wireless nics during active scanning. *International Journal of Information Security*, 7(5):335–348, 2008.

[28] Fabian Lanze, Andriy Panchenko, Benjamin Braatz, and Andreas Zinnen. Clock skew based remote device fingerprinting demystified. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 813–819. IEEE, 2012.

[29] Ding-Jie Huang, Kai-Ting Yang, Chien-Chun Ni, Wei-Chung Teng, Tien-Ruey Hsiang, and Yuh-Jye Lee. Clock skew based client device identification in cloud environments. In *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pages 526–533. IEEE, 2012.

[30] A Selcuk Uluagac, Sakthi V Radhakrishnan, Cherita Corbett, Antony Baca, and Raheem Beyah. A passive technique for fingerprinting wireless devices with wired-side observations. In *Communications and Network Security (CNS), 2013 IEEE Conference on*, pages 305–313. IEEE, 2013.

[31] Ke Gao, Cherita L Corbett, and Raheem A Beyah. A passive approach to wireless device fingerprinting. In *DSN*, pages 383–392, 2010.

[32] Christoph Neumann, Olivier Heen, and Stéphane Onno. An empirical study of passive 802.11 device fingerprinting. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pages 593–602. IEEE, 2012.

[33] Sudip Misra, Ashim Ghosh, Mohammad S Obaidat, et al. Detection of identity-based attacks in wireless sensor networks using signalprints. In *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pages 35–41. IEEE Computer Society, 2010.

[34] Yingying Chen, Jie Yang, Wade Trappe, and Richard P Martin. Detecting and localizing identity-based attacks in wireless and sensor networks. *IEEE Transactions on Vehicular Technology*, 59(5):2418–2434, 2010.

[35] Kai Zeng, Kannan Govindan, and Prasant Mohapatra. Non-cryptographic authentication and identification in wireless networks [security and privacy in emerging wireless networks]. *IEEE Wireless Communications*, 17(5), 2010.

[36] Esrafil Jedari, Zheng Wu, Rashid Rashidzadeh, and Mehrdad Saif. Wi-fi based indoor location positioning employing random forest classifier. In *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, pages 1–5. IEEE, 2015.

[37] Faria and Cheriton. Detection of identity-based attacks in wireless sensor networks using signalprints. In *WiSe '06 Proceedings of the 5th ACM workshop on Wireless security*, pages 43–52. ACM, 2006.

[38] G Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual Design Automation Conference*, pages 9–14. ACM, 2007.

[39] Caidan Zhao, Lianfen Huang, Yifeng Zhao, and Xiaojiang Du. Secure machine-type communications toward lte heterogeneous networks. *IEEE Wireless Communications*, 24(1):82–87, 2017.

[40] Blaise Gassend, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. Silicon physical random functions. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 148–160. ACM, 2002.

[41] Qingqing Chen, György Csaba, Paolo Lugli, Ulf Schlichtmann, and Ulrich Rührmair. The bistable ring puf: A new architecture for strong physical unclonable functions. In *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*, pages 134–141. IEEE, 2011.

[42] Charles Herder, Meng-Day Yu, Farinaz Koushanfar, and Srinivas Devadas. Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE*, 102(8):1126–1141, 2014.

[43] Yingjie Lao, Bo Yuan, Chris H Kim, and Keshab K Parhi. Reliable puf-based local authentication with self-correction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(2):201–213, 2017.

[44] Benjamin W Ramsey, Barry E Mullins, Michael A Temple, and Michael R Grimaila. Wireless intrusion detection and device fingerprinting through preamble manipulation. *IEEE Transactions on Dependable and Secure Computing*, 12(5):585–596, 2015.

[45] Nicholas Kulesza, Benjamin Ramsey, and Barry Mullins. Wireless intrusion detection through preamble manipulation. In *Proceedings of the 9th Int'l Conf on Cyber Warfare and Security*, pages 132–139, 2014.

[46] Xu Jin-yong, Zhao Hang-sheng, and Liang Tao. Method of empirical mode decompositions in radio frequency fingerprint. In *Microwave and Millimeter Wave Technology (ICMMT), 2010 International Conference on*, pages 1275–1278. IEEE, 2010.

[47] Boris Danev, Davide Zanetti, and Srdjan Capkun. On physical-layer identification of wireless devices. *ACM Computing Surveys (CSUR)*, 45(1):6, 2012.

[48] Kasper Bonne Rasmussen and Srdjan Capkun. Implications of radio fingerprinting on the security of sensor networks. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 331–340. IEEE, 2007.

[49] Wenhao Wang, Zhi Sun, Sixu Piao, Bocheng Zhu, and Kui Ren. Wireless physical-layer identification: Modeling and validation. *IEEE Transactions on Information Forensics and Security*, 11(9):2091–2106, 2016.

[50] P Padilla, JL Padilla, and JF Valenzuela-Valdes. Radiofrequency identification of wireless devices based on rf fingerprinting. *Electronics Letters*, 49(22):1409–1410, 2013.

[51] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 454, pages 903–995. The Royal Society, 1998.

[52] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis. Radio frequency fingerprinting for intrusion detection in wireless networks. *IEEE Transactions on Defendable and Secure Computing*, 2005.

[53] Benjamin W Ramsey, Tyler D Stubbs, Barry E Mullins, Michael A Temple, and Mark A Buckner. Wireless infrastructure protection using low-cost radio frequency fingerprinting receivers. *International Journal of Critical Infrastructure Protection*, 8:27–39, 2015.

[54] William E Cobb, Eric W Garcia, Michael A Temple, Rusty O Baldwin, and Yong C Kim. Physical layer identification of embedded devices using rf-dna fingerprinting. In *MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010*, pages 2168–2173. IEEE, 2010.

[55] William E Cobb, Eric D Laspe, Rusty O Baldwin, Michael A Temple, and Yong C Kim. Intrinsic physical-layer authentication of integrated circuits. *IEEE Transactions on Information Forensics and Security*, 7(1):14–24, 2012.

[56] McKay D Williams, Michael A Temple, and Donald R Reising. Augmenting bit-level network security using physical layer rf-dna fingerprinting. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6. IEEE, 2010.

[57] Clay K Dubendorfer, Benjamin W Ramsey, and Michael A Temple. An rf-dna verification process for zigbee networks. In *MILITARY COMMUNICATIONS CONFERENCE, 2012-MILCOM 2012*, pages 1–6. IEEE, 2012.

[58] Komang Oka Saputra, Wei-Chung Teng, and Yi-Hao Chu. A clock skew replication attack detection approach utilizing the resolution of system time. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015 IEEE/WIC/ACM International Conference on*, volume 3, pages 211–214. IEEE, 2015.

[59] Yingying Chen, Konstantinos Kleisouris, Xiaoyan Li, Wade Trappe, and Richard P. Martin. *The Robustness of Localization Algorithms to Signal Strength Attacks: A Comparative Study*, pages 546–563. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[60] Boris Danev, Heinrich Luecken, Srdjan Capkun, and Karim El Defrawy. Attacks on physical-layer identification. In *Proceedings of the third ACM conference on Wireless network security*, pages 89–98. ACM, 2010.

[61] Tektronix arbitrary waveform generators – awg7000 series datasheet. `http://www.tek.com/datasheet/arbitrary-waveform-generators-7`. Accessed: 2017-05-21.

[62] Sheng Wei, James B Wendt, Ani Nahapetian, and Miodrag Potkonjak. Reverse engineering and prevention techniques for physical unclonable functions using side channels. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6. ACM, 2014.

[63] Sheng Wei, Saro Meguerdichian, and Miodrag Potkonjak. Gate-level characterization: foundations and hardware security applications. In *Proceedings of the 47th Design Automation Conference*, pages 222–227. ACM, 2010.

[64] Ali Keshavarzi, Kaushik Roy, and Charles F Hawkins. Intrinsic leakage in low power deep submicron cmos ics. In *Test Conference, 1997. Proceedings., International*, pages 146–155. IEEE, 1997.

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

[66] Javier Galbally, Chris McCool, Julian Fierrez, Sebastien Marcel, and Javier Ortega-Garcia. On the vulnerability of face verification systems to hill-climbing attacks. *Pattern Recognition*, 43(3):1027–1038, 2010.

[67] Ruud M Bolle, Jonathan Connell, Sharath Pankanti, Nalini K Ratha, and Andrew W Senior. *Guide to biometrics.* Springer Science & Business Media, 2013.

[68] David R Raymond and Scott F Midkiff. Denial-of-service in wireless sensor networks: Attacks and defenses. *IEEE Pervasive Computing*, 7(1), 2008.

[69] What are radiotap headers? `http://wifinigel.blogspot.se/2013/11/what-are-radiotap-headers.html`. Accessed=2017-08-11.

[70] Wireshark time stamps. `https://www.wireshark.org/docs/wsug_html_chunked/ChAdvTimestamps.html`. Accessed=2017-08-11.

[71] Nrf52 development kit – product page. `https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52-DK`. Accessed=2017-08-11.