Implications of Radio Fingerprinting on the Security of Sensor Networks

Kasper Bonne Rasmussen
Department of Computer Science
ETH Zurich
8092 Zurich, Switzerland
kasper.rasmussen@inf.ethz.ch

Srdjan Capkun
Department of Computer Science
ETH Zurich
8092 Zurich, Switzerland
srdjan.capkun@inf.ethz.ch

***Abstract*— We demonstrate the feasibility of fingerprinting the radio of wireless sensor nodes (Chipcon 1000 radio, 433MHz). We show that, with this type of devices, a receiver can create device radio fingerprints and subsequently identify origins of messages exchanged between the devices, even if message contents and device identifiers are hidden. We further analyze the implications of device fingerprinting on the security of sensor networking protocols, specifically, we propose two new mechanisms for the detection of wormholes in sensor networks.**

I. Introduction

In recent years, remote device fingerprinting has been successfully performed on a number of devices, ranging from Bluetooth-enabled mobile phones [16] to 802.11 wireless cards [15]. Recently, it has been demonstrated that devices can even be fingerprinted remotely over the Internet [26]. In this paper, we present the first feasibility study of radio fingerprinting of wireless sensor nodes; specifically, we demonstrate that device fingerprinting can be successfully performed on sensor nodes which use Chipcon 1000, 433MHz radios.

The main goal of radio fingerprinting is the detection of signal (device) features that form a valid device fingerprint, based on which associations between observed messages and their senders can be made. In this work, we extracted five signal features and created fingerprints for a set of available sensor nodes; our experimental results show that, based on these fingerprints, an adversary can correctly associate up to 70% of the messages to their respective senders.

So far, device fingerprinting has been mainly studied in the context of privacy violation (e.g., unauthorized user tracking) and device cloning (e.g., detection of SIM-card cloning [35]). However, the implications of device fingerprinting on the security of all-wireless multi-hop (sensor and ad-hoc) networks have been largely neglected. Most sensor networking security protocols were analyzed and designed without taking into account potentially beneficial impact, or harmful consequences of radio fingerprinting. There are two potential benefits of radio fingerprinting in all-wireless (sensor) networks: (i) message authentication and (ii) replay protection[1]. In most security applications strong message authentication is achieved using traditional symmetric-key or public-key primitives. However, in multi-hop networks, nodes can be compromised, replicated [31] and/or can collude through mutually sharing their authentication material. In such a scenario, radio fingerprinting represents an additional form of message authentication that now binds keys and messages to the devices (i.e., to their radio fingerprints). Consequently, radio fingerprinting can be directly used for the prevention of replication [31] and Sybil [10], [29] attacks.

The second important benefit of radio fingerprinting is replay protection. In the context of all-wireless networks, message replay can imply that the message is relayed and replayed to a location which is far from the intended reach of the signal transmitter (i.e., a wormhole [19] is created). Such message replay impacts the nodes' ability to correctly estimate their neighborhood information and fools most distributed topology discovery and localization mechanisms [38]. By binding each key (shared or public) to the device radio fingerprint, replay (and relay) attacks by external attackers can be effectively prevented; this would in turn prevent a large number of attacks on routing and other networking protocols that rely on network topology information. In this work we present a protocol for secure neighborhood discovery using device fingerprinting. We further show how this protocol can be used to prevent wormhole attacks.

As much as it can help to secure sensor networks, device fingerprinting can also be exploited by the attackers to gain information about the network operation or about the users. One example of such an attack consists of detecting and then deactivating (or compromising) the most active network nodes (e.g., cluster heads). Others include the detection of sensing zones (through probing). Recently, a set of key agreement protocols for sensor networks have emerged that rely on device anonymity [6]; radio fingerprinting can severely impact the security of these protocols.

We note here that in this work we analyzed signal

This work was partially supported by the Zurich Information Security Center. This work represents the views of the author.

[1] In Section III-A, we detail the attacker model and assumptions under which radio fingerprinting can be used for authentication and replay protection.

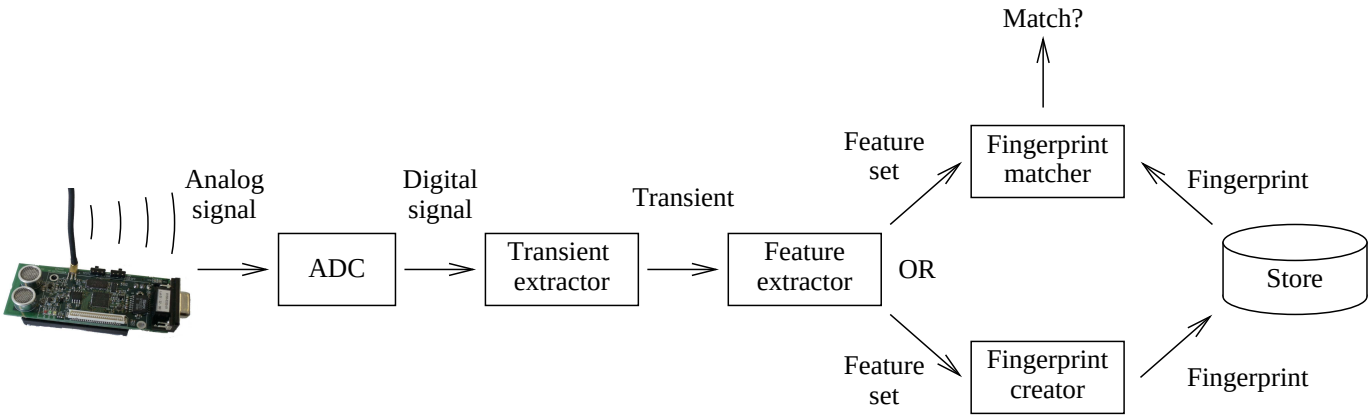

Fig. 1. Radio fingerprinting process.

fingerprints of a set of sensor nodes about which we had no prior knowledge (in terms of how and when they were produced and if they belong to the same or different production lines). We did notice, however, that fingerprints of some pairs of sensors differ significantly, whereas some are very similar. This shows that the success of sensor fingerprinting significantly depends on the choice of sensors by the network authority. If the network authority wants to benefit from sensor fingerprinting (as described above) it can choose sensors for the network whose fingerprints differ significantly and are therefore easy to identify. If, however, the authority wants to prevent unauthorized sensor identification, it can choose sensors whose fingerprints are very similar and hard to distinguish.

In summary, in this paper we make two principal contributions: (i) we demonstrate the feasibility of radio fingerprinting of wireless sensor nodes and (ii) we analyze the implications of radio fingerprinting on the security of sensor networking protocols.

The rest of the paper is organized as follows. Section II details radio fingerprinting and shows our experimental results. In Section III, we analyze the implications of radio fingerprinting on the security of sensor networking protocols. In Section IV, we describe the related work. Finally, we conclude the paper in Section V.

II. Sensor Fingerprinting

In this section we show how, even if sensor communication is anonymized on all upper protocol layers (as described in section III-A), a skilled attacker can use signal fingerprinting [16] to violate sensor anonymity and therefore can associate each individual transmission to a specific node identifier.

We use the terms signal-, radio- and sensor fingerprinting interchangeably throughout the rest of this paper. The different terms are used depending on what layer of abstraction is appropriate for the discussion, but they all refer to the fingerprinting of the signal, sent by the radio, on the sensor.

Sensor fingerprinting is the ability to recognize a specific node based on the analog properties of a signal generated

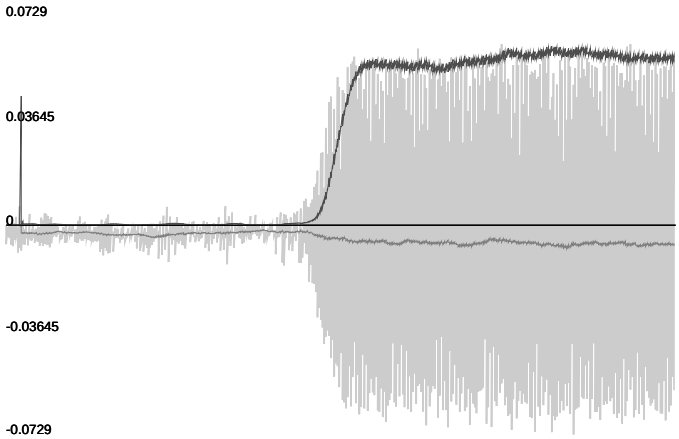

Fig. 2. Radio signal transient. The light gray signal represents the radio signal from the sensor node S, the dark gray signal is the variance signal V whereas the thin medium gray signal is the mean of S over a window of size $w = 50$ i.e., $\overline{X}_{50}$. The scaling constant is $k = 30$

by that node. Our detection scheme is based on the extraction of the radio signal transient and its features. Figure 2 shows the radio signal at the start of a new transmission (or the start of a new packet). The transient is the part of the signal where the amplitude rises from background noise to full power i.e., the central part of Figure 2 (the exact beginning and end of the transient will be defined later in this section). The full radio fingerprinting process is shown on Figure 1. In this process, the radio signal is first received by the fingerprinting device and converted to its digital format. The signal transient is then located and its features are extracted. A set of features form a fingerprint which can later be used for device identification. In this paper the conversion of the signal from analog to digital is done using an oscilloscope. The oscilloscope is used to capture the signal and save a digital version for later processing.

A. Extraction of the transient from the radio signal

The radio signal emitted by a sensor node is captured at the fingerprinting device, composed of an antenna connected to a high-frequency oscilloscope. The analog

signal is then converted by the oscilloscope to its digital version which we denote by S. This digital signal consists of discrete samples; we use S_k to denote the k^{th} sample of signal S.

To aid in the analysis of the signal we define a sliding window of size w. An added benefit of using a sliding window rather than working on the entire signal at once, is the reduced requirements in terms of memory both when the signal is buffered and when the calculations are performed.

The first step in the transient extraction process is the detection of the transient starting point and of its endpoint. The beginning of the signal transient is detected using a threshold detection approach. In [16] Hall et al. have demonstrated that a threshold detection approach based on variance can be applied to successfully identify the start of the transient. We use the similar approach in our work. To detect the start of the transient we define a new discrete variance signal V as:

$$V_i = k\frac{1}{w-1}\sum_{n=1}^{w}(S_{i-n} - \overline{X}_w)$$

where $\overline{X}_w$ is the mean of the values $S_{i-w}, \ldots, S_{i-1}$, w is the sliding window size and k is an appropriate scaling constant. The scaling constant k is used to make the values of the V signal comparable to those of the measured sensor radio signal S. The variance signal V can be viewed as a measure of how much the incoming signal S deviates from the average values of the last w samples. When the transient occurs, S will rapidly increase causing the deviation to be high and thus detecting the transient. An example of a measured radio signal S and its variance signal V (scaled by $k = 30$) is shown in Figure 2.

Once the signal is detected and its variance computed, the problem of finding the start and end of the transient is now reduced to a change-point detection problem. For the start of the transient the change-point is the point where the V signal starts to rise, and for the end of the transient it is the point where the V signal flattens out. We have solved these two change-point detection problems using the CUSUM algorithm [27].

The CUSUM algorithm is short for cumulative sum and it works by adding the latest increase (or decrease) in the V signal to the sum, and then subtracting a fixed amount. This means that only when the V signal is climbing faster than a predefined rate will the cusum of the signal rise. More precisely, the cusum of the signal is defined as follows:

$$\text{CUSUM}(V_i) = \max\big(\text{CUSUM}(V_{i-1}) + (V_i - V_{i-1}) - \alpha, 0\big)$$

By definition, $\text{CUSUM}(0) = 0$.

We now further define the detection signal $D = \text{CUSUM}(V)$. This signal will only begin to rise when the signal variance V rises significantly (i.e., when the start of the transient is detected. The detection signal falls back to zero when the V signal flattens out.

The start of the transient is defined as the point at which the detection signal D rises above a predefined threshold t.

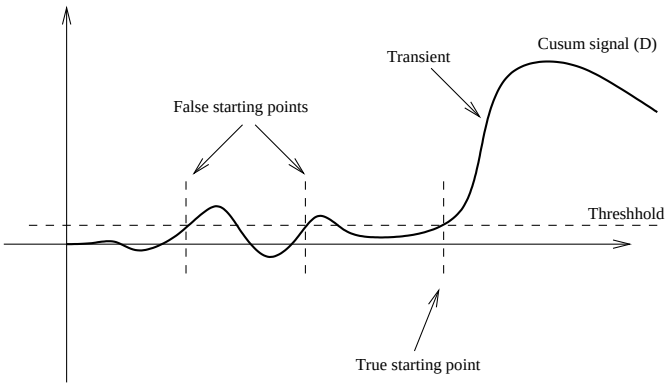

Fig. 3. An illustration of the detection of the starting point of the signal transient.

If t is set too low the detection scheme is more susceptible to noise and if it is set to high the detection of the start of the transient will be delayed. In Figure 3 is an illustration of how the threshold can influence the detection of the start of the transient. We have found that threshold values between $t = 1 \cdot 10^{-5}$ and $t = 5 \cdot 10^{-5}$ works best depending on the amount of noise on the channel and the amplitude of the signal. If the threshold is set too low, there is a possibility that false starting points will be detected. We have solved this problem by deferring the decision about which starting point is the right one untill we detect the endpoint of the transient. When the endpoint is detected, the starting point is chosen as the last point at which D raises above the threshold t (see Figure 3). The value of the threshold is still important. The decision of whether we are seeing a new possible start of the transient or just a dip in the D signal is determined by whether the D signal drops below the threshold (i.e., if the threshold is set too low, the start of the transient will be detected well before the actual transient starts).

In order to find the transient endpoint we use a look-ahead-window of length ℓ to see if the signal D has reached its peek. The end of the transient is found at the point at which the following equation becomes true:

$$D_i \geq \max(\{D_{i+1}, \ldots, D_{i+\ell}\})$$

When we in the following refer to the length of the transient we mean the distance from the transient starting point to its endpoint along the x-axis.

B. Extract features from the transient

In order to differentiate between different transients (i.e., recognize different nodes) we need to look at specific characteristics of this signal; we call these characteristics transient *features*. Different features have been suggested in the open literature. Hall et al. [16] suggested 10 different signal features, using both signal frequency and amplitude information. Ellis and Serinken [11] also includes both amplitude and phase information in their fingerprints of VHF radios. Ellis and Serinken found, however, that even with all extracted features, some radios are virtually

indistinguishable. This suggests that the feature set which will yield the best fingerprint is highly dependent on the type of radio being fingerprinted.

We are fingerprinting the small CC1000 radio on a "Cricket" node and because we sample the signal at particular intervals we get values representing signal strength (amplitude) i.e., without any readily available frequency information. Because we have no frequency information we limit ourselves to features that are based on what we can directly measure, i.e., the (relative) signal amplitude. We note that it is possible to estimate the signal phase using techniques described in [17]. In this work, however, we limit our study to the features derived from signal amplitude; extracting features using estimated signal phase is a possible topic of our future research.

A good signal feature has a low intra radio variability (from sample to sample in the same radio) but a high inter radio variability (between different radios). Using features with those properties will ensure a good stability and will provide a good way to distinguish between the radios. It has however been our experience that a feature that stays stable from sample to sample in the same radio does not change much when measured on a different radio. The opposite is also true.

We observed the following signal features:

1) The length of the transient, along the x-axis. (len)
2) The variance of the normalized amplitude of the transient. (var)
3) The number of peaks (periods) of the carrier signal in the transient. ($peaks$)
4) The first part of a discrete wavelet transform [3] of the transient. ($dwt0$)
5) Difference between the normalized mean and the normalized maximum value of the transient. ($ndif$)

Before we go into which features perform better than others we will take a look at the fingerprints and how they are created. A fingerprint must fulfill a number of criteria. First of all it must be of a reasonable size. The size of a fingerprint is limited by the available storage on the fingerprinting devices; this is especially important if the fingerprinting device is a mote-class [2] sensor node. The fingerprint should also preserve as much relevant information about the signals used to create it as possible. There is therefore an obvious tradeoff between retaining information within the fingerprint and its size.

We represent the fingerprint as a vector of average feature values μ and a covariance matrix C. The size of the fingerprint can be varied with the number of different features in contains. We calculate the probability of a signal match to the stored fingerprint using a Kalman filter and a technique proposed by Bar-Shalom [40].

$$P(\bar{u}) = exp\left(-\frac{1}{2}(\bar{u}-\mu)^\top C^{-1}(\bar{u}-\mu)\right)$$

where $\bar{u}$ is a vector of the feature set of the signal we want to check.

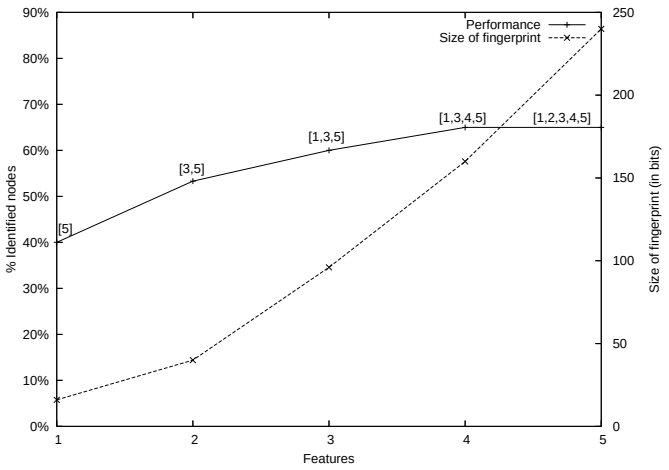

Fig. 4. Number of correctly identified beacons and the size of the fingerprint as a function of the number of features (averaged over 6 tries). The numbers in the square brackets next to the data points on the performance graph represent features that are included in that fingerprint. 1=len, 2=var, 3=peeks, 4=dmt0, 5=ndif

C. Selection of features

As we discussed in the previous section, the size and the quality of the fingerprint can be altered depending on the number and the type of features included in the fingerprint. In order to better understand how different features influence the fingerprinting process, we create 6 data sets for each sensor node. Each data set consists of 100 samples; 50 samples from which the fingerprint is created, and 50 samples representing the signal to match against.

A beacon is correctly identified if the fingerprint created from the first half of the data set (i.e., the first 50 samples) result in a match when tested against the signal data from the last half of the data set. The graph in Figure 4 shows the average percentage of correctly identified beacons (averaged over 6 tests) as a function of the number of features in the fingerprint. Note the performance also increases if we include more *samples* in the fingerprint and test data. This graph is based on the best feature for the 1-feature data point, the best combination of 2 features for the 2-feature data point, the best combination of 3 features for the 3-feature data point etc. Figure 4 also shows the size of a fingerprint as a function of the number of features.

It is interesting to note that the best combination of features for an n-feature fingerprint is a subset of the best combination for a $(n+1)$-feature fingerprint. This suggests that it is the quality of the features, and not their combination, that ultimately determines the quality of the fingerprint.

D. Performance evaluation

From these results, we can observe that the sensor identification rate is around 70%. This means that the sensors will be correctly identified in 70% of cases. In the the remaining 30% of the cases, the detection will fail due to interference or environment variation. This, however,

Feature Set	Node Numbers #1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Sum
[1,2,3,4]	6	2	3	2	4	1	5	1	3	2	27
[1,2,3,5]	6	2	6	6	3	4	4	2	3	2	38
[1,2,4,5]	6	3	6	6	2	2	3	3	3	2	35
[1,3,4,5]	6	3	6	6	3	3	4	1	5	2	38
[2,3,4,5]	6	3	6	6	2	2	3	3	4	2	36
Sum	30	13	27	26	14	12	19	11	18	10	

TABLE I

THE TOTAL NUMBER OF SUCCESSFUL IDENTIFICATIONS, IN 6 TRIES, FOR EACH 4-FEATURE FINGERPRINT (FINGERPRINT SIZE = 50 SAMPLES)

does not mean that the attacker will be able to successfully pose as another node (e.g., an attacker, node #1, tries to pretend to be node #3) in 30% of the cases. If we assume that when the identification fails, another random node will be identified as the sender, there will be a 1 in $N-2$ chance that the node that is (erroneously) identified is the exact one that the attacker was impersonating.

If there are 10 nodes in total, the probability of an attacker successfully forging its identity is reduced to:

$$30\% \cdot \frac{1}{8} \approx 3.75\%$$

If a node can successfully clam to be #1 3 times there is only a 0.005% chance that it is lying about its identity.

Feature quality: It is clear that all the features do not have an equal impact on the quality of the fingerprint. Table I lists ten nodes and all possible combinations of features in a 4-feature fingerprint. All the fingerprints have been tested with 6 different datasets for each of the ten nodes. The numbers in the table counts the number of times that the fingerprint correctly identified the node i.e., the highest number possible is 6. If we look at node #3; it is identified correctly in all cases except in the one where feature 5 is missing (the first fingerprint), the same is true for node #4. In fact the first fingerprint performs consistently bad compared to the rest, a fact that can also be confirmed by the sum in the right most column. The observation that feature number 5 is probably one of the better features is further confirmed by Figure 4 since it is the only feature that is present in all the fingerprints.

It is not only the features that influence the performance on the fingerprinting framework. If we look at node #8 we can see that it has only been correctly identified a total of 11 times (out of 30) whereas node #1 has been correctly identified every time, by *all* the fingerprints. This shows clearly that some devices (such as node #1) have characteristics that are more easily identified (because they are more extreme). More easily identifiable devices can be selected specifically in order to create a sensor network with a better detection probability.

Even though the performance curve in Figure 4 is leveling out at approximately. 70%, it is well possible that the performance can be improved with a different set of features and with carefully choosing nodes in the network such that their fingerprints sufficiently differ; this, we leave for future work.

E. Experimental setup

This section provides details about the equipment and methods used to generate and capture the signals from the sensors.

433MHz radio signals are generated by MIT's Cricket (sensor motes with CC1000 radios [1]). Since we are only interested in the signal in and around the transient, the Crickets transmit one pulse (i.e., one TinyOS packet) every second. The transmissions are picked up by an antenna tuned to 433MHz connected to a high frequency oscilloscope. The antenna is connected directly to the oscilloscope to avoid any interference from LC–filters and other circuitry on the receiver. The signal is sampled at 1GHz and stored by the oscilloscope for processing. 10 Cricket motes of model *MCS410CA, Cricket 232* were used and 200 signals were collected from each Cricket resulting in a test base of 2000 signals. All the subsequent signal processing is done on a standard desktop PC. However the algorithms are written in plain C to simplify porting them to the motes in future applications.

III. IMPLICATIONS OF RADIO FINGERPRINTING ON THE SECURITY OF SENSOR NETWORKS

In this section we analyze the implications of radio fingerprinting on the security of various protocols. As we showed in the previous sections, it is possible to determine if two sets of transmissions of 30 (or more) packets each, originate from the same node. Although it might be possible to disrupt the fingerprinting procedure (as we will explain in section III-E), we assume for now that a reasonable degree of detection can be performed and we will look at how that affects the security of a subset of sensor networking protocols. In the following we further assume that the fingerprint can not be forged by the attacker.

We first describe our system and attacker models.

A. System and Attacker model

Our system consists of a network of sensor nodes that communicate via radio transmissions. The network is operated by an authority. This authority can be on-line, meaning that the authority operates on-line servers (by single hop or multi-hop communication), or off-line, meaning that the services of the authority cannot be reached via the network. In either case, the authority controls the network membership and assigns a unique identity to each

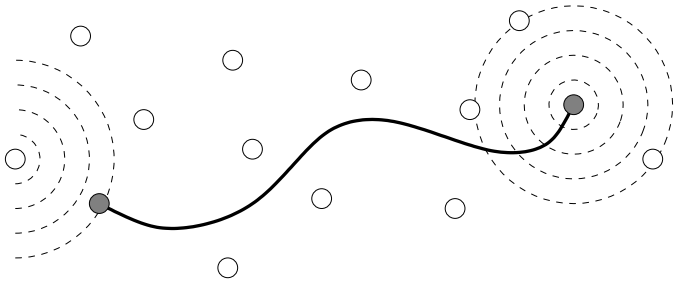

Fig. 5. A Wormhole. This Figure illustrates two colluding external attackers who have formed a fast wormhole using an alternate connection.

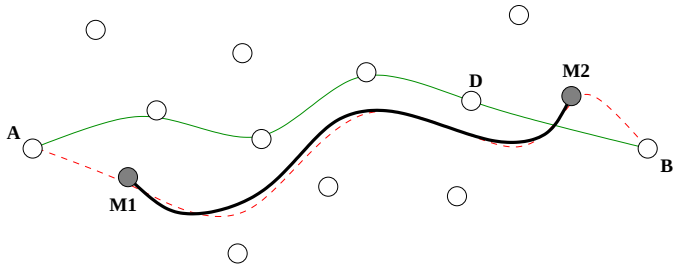

Fig. 6. An illustration of a wormhole attack. $M1$ and $M2$ are colluding external attackers that have formed a wormhole from (next to) the source A to (next to) the destination B. Using fingerprinting B is able to detect the wormhole since $M1$ does not have a fingerprint corresponding to A (or to another legitimate network node).

node. We assume that all network nodes can establish pairwise secret keys. This can be achieved by manually pre-loading all keys into the nodes in a network setup phase, by probabilistic key pre-distribution schemes [12], [7], or through an on-line key distribution center [20].

We adopt the following attacker model. We assume that the attacker controls a set of verification devices that are equipped and located such that they can seamlessly observe all communication between sensor nodes (e.g., verifiers can be equipped with high gain antennas and located in the proximity of the network). The attackers are not a part of the network controlled by the authority and cannot gain access to network keys or disclose any messages exchanged between the sensor nodes (or between the nodes and the authority). We do assume that the verifiers can detect and separate radio signals originating from sensor nodes. This means that the verifiers can detect that a transmission is taking place and separate the transmission from other signals and noise. We further assume that, as we showed in the previous sections, an attacker can detect if two sets of messages (≥ 30 packets each) originated from the same device.

B. Detection of wormhole attacks

In this section we show how sensor fingerprinting can be used for the detection of external replay and wormhole attacks. We propose two different wormhole detection methods, a centralized method that rely on the infrastructure to issue a warning if any wormholes are detected and a decentralized method in which all nodes have fingerprinting capabilities. We start with a brief description of what a wormhole is, and why it is difficult to detect a wormhole attack.

1) Wormholes attacks: In a wormhole attack [19], an attacker receives packets at one point in the network, tunnels them to another point in the network, and then replays them into the network from that point. For tunneled distances longer than one hop it is simple for the attacker to make the tunneled packet arrive sooner than other packets transmitted over a normal multihop route (e.g., through use of a single long-range directional wireless link or through a direct wired link to the colluding node). It is also possible for the attacker to forward each bit of the packet over the wormhole directly, without waiting for an entire packet to be received, in order to minimize the delay introduced by the wormhole. Due to the nature of wireless transmission, the attacker can create a wormhole even for packets not addressed to itself, since it can overhear them on the wireless channel and tunnel them to the colluding node at the other end of the wormhole. Figure 5 illustrates a wormhole formed by two external attackers.

A wormhole attack is among the most difficult attacks to detect because it can be executed exclusively by external attackers and because the information in the packets does not need to be changed for the attack to work. That means that even encrypted and signed messages can be subject to a wormhole attack.

If a wormhole attack is carried out against a routing protocol such as AODV [33] or DSR [22] RouteRequest packets can be tunneled through the wormhole directly to the target of the Request. When the neighbors of the destination node hears the Request they will rebroadcast that Request and then discard all subsequent RouteRequest packages from the same route discovery. The consequence of this is that the only route from the source to the destination will be through the wormhole which leaves the attacker in a unique position to filter out unwanted packets or simply refuse to forward traffic.

Several methods [5], [34], [37] have been suggested to deal with this kind of attack. Hu, Perrig and Johnson [19] suggested packet leashes and the TIK protocol which attempts to upper bound the time it takes to send a message from one place to another. Packet leashes, however, are not particularly suited for sensor networks because the TIK protocol requires very tight time synchronization. Hu et. al. suggests that GPS is used as a source of precise time but that is not always an option in low cost sensors.

2) Centralized detection of wormhole attacks: Our proposal is to use the fingerprinting techniques described in section II to allow the network authority to detect wormhole attacks. Since a node has little or no control over its fingerprint it is a factor that will differ between the original sender of a message and any node that tries to replay it. We propose an infrastructure (i.e., one or more central nodes) controlled by an authority, equiped with an oscilloscope capable of sampling the signal and creating a fingerprint (see Figure 7). The infrastructure must also be

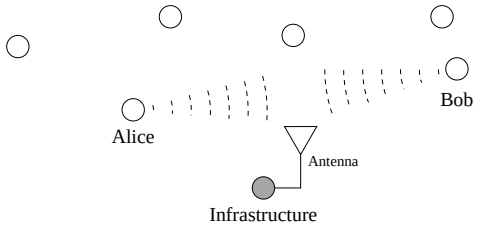

Fig. 7. Centralized detection of wormhole attacks. The infrastructure is listening passively to all transmissions. If there is an identity mismatch between the key used to sign the message and the fingerprint, the infrastructure will issue a warning.

```
1 when new_message_arrives
2     id ← fingerprint()
3     msg ← receive(n‖payload‖MAC_{K_n})
4     if integrity(msg) = OK
5         if id ≠ n
6             broadcast_alert()
```

Fig. 8. Pseudo code of how the centralized wormhole detection mechanism works. For every message, the infrastructure first checks the message integrity and then checks if the key used to create the MAC of the message, corresponds to the identity associated with the signal fingerprint.

capable of receiving all the relevant signals on the wireless channel i.e., only the area covered by the infrastructure is protected from wormholes. Figure 8 explains in pseudo code how the centralized wormhole detection mechanism works. First the identity of the sender is determined based on the fingerprint of the signal. After the message is received the authority checks the integrity of the message (i.e., generate a MAC of the message using key K_n and check if it corresponds to the MAC of the message). This is done for two reasons; the first is to verify the identity of the sender, and the second is to avoid sending out an alert if the message was corrupted in transit (e.g., by jamming or interference) since the nodes will detect this on their own. If the integrity is verified but the identity n of the message does not correspond to the identity id of the fingerprint then an alert will be broadcast.

In order for the infrastructure to be able to link a specific identity to a fingerprint, the network authority must know the reference fingerprints of all legitimate nodes in the network. The network authority must also know the keys of all the legitimate nodes in the network in order to verify the integrity of the message. The keys can be, e.g., public keys or shared keys that the network authority have confirmed before deployment, or there can be a registration procedure in which a new node can register a key, by sending multiple signed packets to the infrastructure, thus allowing the network authority to create a good reference fingerprint.

3) Distributed Detection of wormhole attacks: Part of out future work includes implementing the detection mechanism directly in the nodes so they can check the fingerprint of any incoming transmission themselves. On Figure 6 we show an example of an (attempted) wormhole attack in a scenario where all the nodes have fingerprinting capabilities. Node B will be able to detect the wormhole attack since the signature of the message corresponds to node A but the fingerprint of the signal will be that of node $M2$. In the situation where the message have been routed through the network the signature will claim D is the sender, which the fingerprint will confirm.

One limitation of this approach is that it cannot detect the wormhole attack if the node nearest the destination is an internal attacker. In that case the internal attacker can change the data in the message or wrap its own identity around the original data, pretending to forward the message, and retransmit it with a matching identity and fingerprint.

In the distributed scenario, the detection process requires that the receiving node knows the reference fingerprint of the sender. The simplest way to ensure that the receiving node has the reference fingerprint of the sender is to pre-load the reference fingerprints of all the nodes onto every node before deployment. In a network of, say, 10.000 nodes and a fingerprint size of 30 bytes (240 bits) it would take about 300kB of storage on each node to store all the fingerprints, which is not prohibitively expensive. Further more the fingerprints would not have to be stored in main memory they could be stored in a separate flash-ROM so as to not take space from the normal functionality of the node.

If all fingerprints are preloaded in all nodes we need to solve the problem of adding new nodes to an already existing wireless sensor network. It can be done e.g., by an authenticated broadcast from a central authority that contains the fingerprints of the newly added nodes, or a new node could just inform the others about it's fingerprint. If a new node just needs to broadcast its fingerprint to join an established sensor network, it is of course possible for the new node to send a false fingerprint, however, sending a false fingerprint will only exclude it from the sensor network, since other nodes will then discard all future packets from that node as false.

We propose a secure neighborhood discovery protocol to combat wormhole attacks and to enable secure neighborhood discovery. The protocol can be seen in Figure 9. It uses a combination of fingerprints and MACs to archive both entity authentication and message authentication. A protocol with MACs as the only protection mechanism could verify the key of the other node, but would not be able to detect a clone. A protocol with fingerprints alone would be able to verify the identity of the other node (which might be enough in some scenarios), but would not be able to detect if the message was compromised e.g., partially jammed by the attacker.

First Alice picks a k bit random nonce N_A and broadcasts it, along with her identity, to start the neighborhood discovery process. Upon receiving this Bob compares the fingerprint of the signal to his reference fingerprint of Alice. If these two fingerprints do not match Bob will ignore the message. This step has the added benefit of reducing the

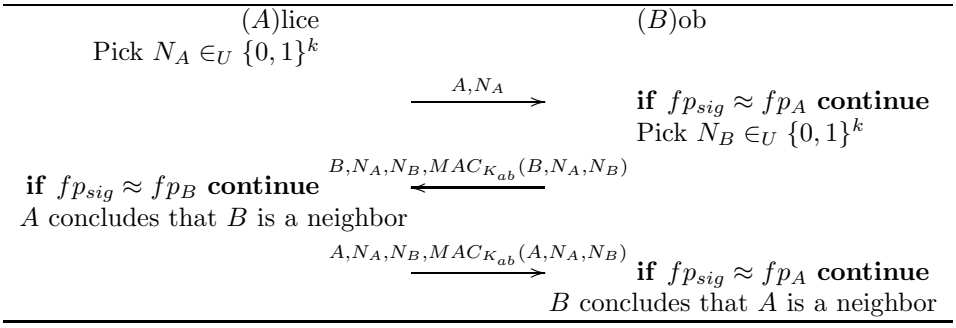

Fig. 9. Neighborhood discovery protocol with fingerprint identification. After a successful run of this protocol A and B both conclude that they are one-hop neighbors.

effectiveness of battery draining attacks since Bob does not have to reply even if the sender is an Alice-clone. If the fingerprints do match, Bob picks a random nonce N_B and sends it back along with Alice's nonce and a MAC to protect the message. If the fingerprint of that signal matches the reference fingerprint of B Alice sends the nonce N_B back to Bob who again has a chance to verify the fingerprint and the cryptographical identity of Alice (i.e., verify that she holds the key K_{ab}).

If this protocol is successfully executed Alice and Bob will establish each others identities and will be assured that they reside in each others direct communication range (i.e., they are one hop neighbors).

C. Detection of Sybil and Cloning Attacks

Another straightforward benefit of radio fingerprinting in sensor networks is the detection of Sybil [10] and replication [31] attacks. In the Sybil attack, the attacker assigns several identities to the same network node. These identities can be fake, but they might also be true identities of nodes that the attacker compromised. The replication attack consists of assigning the same (legitimate) identity to several nodes. All the concerned nodes are under the same attackers control, and are all assigned the identity of a legitimate network node under the attackers control.

Both attacks can be prevented, either using a variation of the protocol in Figure 9 or using the centralized detection method described in section III-B.2.

With Sybil and cloning attacks prevented, the impact of node compromise on the execution of network protocols is significantly reduced, as the compromised nodes can now only harm protocols locally. If these attacks are not prevented, a compromised node can be replicated by the attacker (i.e., all node authentication material is copied to the nodes controlled by the attacker); these replicas can then be placed in different network neighborhoods and collude to prevent the correct execution of network protocols, including routing, localization, time synchronization and distributed data aggregation. With radio fingerprinting, such attacks can be effectively prevented.

We acknowledge that specific techniques for the detection of Sybil and cloning attacks were also proposed that do not rely on radio fingerprinting [29], [31]. Still, these techniques tackle cloning and/or Sybil attacks only, whereas the benefits of device fingerprinting are much broader.

D. The misuse of sensor fingerprinting

In the previous sections we saw examples of how sensor fingerprinting can be used by the network authority to detect attacks. In this section we explore the possibilities of an attacker armed with sensor fingerprinting capabilities.

Most of the possibilities for misuse center around tracking and identification of nodes. One example of such an attack consists of detecting and then deactivating (or compromising) the most active network nodes (e.g., cluster heads). Other examples include the detection of sensing zones (through probing). Although fingerprinting by an external attacker can be partially addressed with the use of spread-spectrum communication [39] with secret spreading codes, it is worth noting that with most current sensor platforms this protection is not implemented.

1) Attacks on Key Establishment: Recently, a key establishment protocol for sensor networks has emerged that relies on device anonymity. In [6] Castelluccia and Mutaf propose a protocol called "Shake 'em up!", in which a secret key is established between two (sensor) nodes that share no prior secrets or credentials. In this protocol, Alice can send the secret bit 1 to Bob by broadcasting an (empty) packet with the source field set to Alice i.e., the actual sender. Similarly, Alice can send the secret bit 0 to Bob by broadcasting an (empty) packet with the source field set to Bob.

Only Bob can identify the real source of the packet (since it did not send it, the source is Alice), and can recover the secret bit (1 if the source is set to Alice or 0 if the source is Bob). An eavesdropper cannot retrieve the secret bit since it cannot figure out whether the packet was actually sent by Alice or Bob. By randomly generating n such packets Alice and Bob can agree on an n-bit secret key. This protocol therefore exclusively relies on the fact that the attacker cannot detect from which node the packets were sent. The authors of this paper do address device identification through RSSI measurements and thus suggest that the nodes are shaken-up to prevent such attacks. However, our initial investigations indicate that

sensor fingerprinting is more robust to such limited device motion. It is therefore well possible that a skilled attacker can brake this protocol using radio fingerprinting.

E. Robustness of fingerprinting

One of the assumptions when performing radio fingerprinting is that the fingerprinting device is able to separate the signals from the different nodes participating in the communication. This assumption however will not always be true. An attacker who transmits a weak jamming signal will be able to alter the signal characteristics enough to prevent the authority from accurately identifying the nodes.

A similar principle can be used as a protective measure. Namely, a network authority can generate a signal that would change the characteristics of transmissions within the network enough to confuse the attacker, but without destroying the content of the transmissions, thus by preventing device fingerprinting. This is a topic of our future research.

IV. Related work

Signal detection and identification have been a topic of interest since the early development of radar systems around the time of World War II [23]. The problem of identifying the source of a transmission is still to this day a problem that researchers are focusing on; e.g., mobile phone operators have addressed this problem in an attempt to combat cell phone cloning [28], [30], [35].

Signal fingerprinting is relatively new within the world of sensor networks. J. Hall, M. Barbeau and E. Kranakis [16] have published work on identifying the transient in Bluetooth devices and have been a source of inspiration for our work, however their work focuses on the use of signal phase where we use other parameters in our detection scheme (see Section II).

While methods for characterizing radio transmitters have been suggested for both transient analysis [4], [11], [16], [21], [36], [13], [14], [8] as in our case, and frequency based identification using wavelets [18], [9], [24], [25], [32] none so far have applied these techniques to the identification of nodes in a sensor network.

A number of researchers have also explored device fingerprinting for wired devices. Kohno, Broido and Claffy [26] use a feature of the TCP or ICMP protocol[2] to estimate the clock skew of a particular device and are able to create fingerprints based on that.

V. Conclusion

In this work we demonstrated the feasibility of radio fingerprinting of wireless sensor nodes (Chipcon 1000, 433MHz radios). In our experiments, we were able to create radio fingerprints and subsequently identify origins of messages exchanged between the devices, even if message contents and device identifiers were hidden. We analyze the implications of radio fingerprinting on the security of several sensor networking protocols and attacks. We show that device fingerprinting, so far mostly neglected in analysis of sensor network security, can be both beneficial as well as harmful for the security of sensor networks.

In this paper, we only scratched the surface of radio fingerprinting and its implications on wireless sensor networks. A number of issues are still left open in this investigation, from the formations of better fingerprints to the impact of noise and mobility on the fingerprinting process.

[2]The TCP timestamping option from RFC 1323 is implemented by most modern TCP stacks and ICMP timestamp requests (ICMP message type 13)

References

[1] The cricket indoor location system. http://cricket.csail.mit.edu/.

[2] Mica sensor platform. http://www.xbow.com.

[3] Wikipedia – discrete wavelet transform. http://en.wikipedia.org/wiki/Discrete_wavelet_transform.

[4] M. Barbeau, J. Hall, and E. Kranakis. Intrusion detection and radio frequency fingerprinting in mobile and wireless networks. Technical report, Carleton University, School of Computer Science, October 2003.

[5] Levente Buttyán, László Dóra, and István Vajda. Statistical wormhole detection in sensor networks. In *ESAS*, pages 128–141, 2005.

[6] Claude Castelluccia and Pars Mutaf. Shake them up!: a movement-based pairing protocol for cpu-constrained devices. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 51–64, New York, NY, USA, 2005. ACM Press.

[7] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, page 197. IEEE Computer Society, 2003.

[8] D. L. Mensa et al. Radar signature evaluation apparatus, March 2003. United States Patent 6,529,157.

[9] D. N. Hoogerwerf et al. Active waveform collection for use in transmitter identification, March 2000. United States Patent 6,035,188.

[10] J. Douceur. The Sybil Attack. In *Proceedings of the IPTPS02 Workshop*, Cambridge, MA (USA), 2002.

[11] K. J. Ellis and N. Serinken. Characteristics of radio transmitter fingerprints, 2001.

[12] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47, New York, NY, USA, 2002. ACM Press.

[13] P. J. Ferrell. Method and apparatus for characterizing a radio transmitter, April 1991. United States Patent 5,005,210.

[14] M. B. Frederick. Cellular telephone anti-fraud system, September 1995. United States Patent 5,448,760.

[15] Ryan M. Gerdes, Thomas E. Daniels, Mani Mina, and Steve F. Russel. Device identification via analog signal fingerprinting: A matched filter approach. *The 13th Annual Network and Distributed System Security Symposium*, 2006.

[16] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis. Detection of transient in radio frequency fingerprinting using signal phase. *Wireless and Optical Communications*, 2003.

[17] Jesse Hansen. Selected approaches to estimation of signal phase. Technical report, University of Rhode Island, 2003.

[18] R. D. Hippenstiel and Y. Payal. Wavelet based transmitter identification. In *The Fourth International Symposium on Signal Processing and Its Applications (ISSPA 96)*, Gold Coast, Australia, 1996.

[19] Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks. In *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, San Francisco, USA, April 2003.

[20] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne:: a secure on-demand routing protocol for ad hoc networks. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 12–23, New York, NY, USA, 2002. ACM Press.

[21] J. Hall and M. Barbeau and E. Kranakis. Detection of transient in radio frequency fingerprinting using signal phase. Slide Presentation, October 2003.

[22] D. Johnson, D. Maltz, and J. Broch. *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[23] R. Jones. *Most Secret War*. Hamilton, 1978.

[24] J. Kamarainen, V. Kyrki, and T. Lindh. Signal discrimination based on power spectrum of filter response. Research Report 80, Lappeenranta University of Technology, Department of Information Technology, 2002.

[25] D. Kaplan and D. M. Stanhope. Waveform collection for use in wireless telephone identification, December 1999. United States Patent 5,999,806.

[26] Tadayoshi Kohno, Andre Broido, and K. C. Claffy. Remote physical device fingerprinting. *IEEE Trans. Dependable Secur. Comput.*, 2(2):93–108, 2005.

[27] George V. Moustakides. Performance of cusum tests for detecting changes in continuous time processes. In *2002 IEEE International Symposium on Information Theory*, pages 186–, 2002.

[28] H. Mustafa, M. Doroslovacki, and H. Deng. Automatic radio station detection by clustering power spectrum components. In *The IEEE International Conferenceon Acoustics, Speech, and Signal Processing, (ICASSP 02)*, volume 4. IEEE, May 2002.

[29] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses, 2004.

[30] T. L. Overman and K. C. Overman. Adaptive radar threat detection and tracker verification system, March 1979. United States Patent 4,146,892.

[31] Bryan Parno, Adrian Perrig, and Virgil Gligor. Distributed detection of node replication attacks in sensor networks. In *Proceedings of IEEE Symposium on Security and Privacy*, May 2005.

[32] Y. Payal. Identification of push-to-talk transmitters using wavelets. Master's thesis, Naval Postgraduate School, Monterey, CA, 1995.

[33] C. Perkins, E. M. Royer, and S. R. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. IETF Internet draft. July 2000.

[34] Radha Poovendran and Loukas Lazos. A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. *ACM Journal on Wireless Networks (WINET)*, 2005.

[35] D. Shaw and W. Kinsner. Multifractal modeling of radio transmitter transients for classification. In *The IEEE Conference on Communications, Power, and Computing (WESCANEX 97)*, page 306?312, Winnipeg, Manitoba, Canada, May 1997. IEEE.

[36] O. Ureten and N. Serinken. Detection of radio transmitter turn-on transients. Electronics Letters, November 1999.

[37] S. Čapkun, L. Buttyán, and J.-P. Hubaux. SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, Washington, USA, October 2003.

[38] S. Čapkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, 2005.

[39] Andrew J. Viterbi. *CDMA. Principles of Spread Spectrum Communication*. Addison Wesley Longman, Inc., 1995.

[40] Thiagalingam Kirubarajan Yaakov Bar-Shalom, X.-Rong Li. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2002.