Spatial Signatures for Lightweight Security in Wireless Sensor Networks

Lifeng Sang and Anish Arora
Department of Computer Science and Engineering
The Ohio State University, Columbus, Ohio 43210
Email: {sangl, anish}@cse.ohio-state.edu

***Abstract*— This paper experimentally investigates the feasibility of crypto-free communications in resource-constrained wireless sensor networks. We exploit the spatial signature induced by the radio communications of a node on its neighboring nodes. We design a primitive that robustly and efficiently realizes this concept, even at the level of individual packets and when the network is relatively sparse. Using this primitive, we design a protocol that robustly and efficiently validates the authenticity of the source of messages: authentic messages incur no communication overhead whereas masqueraded communications are detected cooperatively by the neighboring nodes. The protocol enables lightweight collusion-resistant methods for broadcast authentication, unicast authentication, non-repudiation and integrity of communication. We have implemented our primitive and protocol, and quantified the high-level of accuracy of the protocol via testbed experiments with *CC1000* radio-enabled motes and *802.15.4* radio-enabled motes.**

I. Introduction

Authenticity of information is critical to wireless sensor applications. In event detection, for instance, a message may bring critical information about a particular region. Event handlers would need assurance that the location information in the message is authentic and that its content has not been modified. They may even wish to reconfirm the occurrence of the event. Scenarios like this motivate the need for properties such as broadcast/unicast message authentication, integrity, and non-repudiation. In essence, the need is for an efficient basis for one-hop message authentication, as hop-by-hop security is typically preferred when resources are constrained. We envision that the need for such security properties will only grow as applications start dealing with control scenarios.

The conventional approach to message authentication relies on using secrets. However, cryptography with even symmetric secrets can consume significant overhead in wireless sensor networks, especially low power ones. Other complications include the ease of eavesdropping given the broadcast nature of the medium, which makes applications vulnerable to malicious behavior. Moreover, the potentially large number and dynamic nature of nodes pose a key management challenge [10].

These challenges lead us to investigate the feasibility of crypto-free communications in resource-constrained wireless sensor networks. Towards establishing trust among a set of nodes without using secrets, we turn towards exploiting physical features of nodes that have the potential for being unique.

The specific concept we propose is that of the "spatial signature" of a node, which is a physical characterization of the signal that the node induces at each of its neighbors. In this paper, we show experimentally that a spatial signature of nodes based on physical features such as Received Signal Strength Indicator (RSSI) or Link Quality Indicator (LQI) is unique with high probability, in multiple radio platforms and in diverse network topologies that range from rather sparse to very dense. It also enjoys desirable properties of stability and ease of learning. We are thus able to design a lightweight and robust primitive that validates the spatial signature of messages at run-time. The primitive, being statistical in nature, can produce both false positives and false negatives; our experiments however show that we can efficiently instrument it so that there are no false positive and rare false negatives in diverse networks. The memory and latency requirements of our primitive are substantially less than those of extant secret processing methods in wireless sensor networks.

Based on the primitive, we design a cooperative

protocol that uses the primitive to perform message source authentication. The central idea of our cooperative protocol is this: a succinct representation of the spatial signature induced by a node on its neighbor is stored at the neighbor. If the adversary sends a message masquerading as the node, a spatial signature anomaly is detected and reported by the intended receiver(s) of the message, some neighbors of the node, and/or some neighbors of the adversary. Conversely, if a message is authentic, the spatial signature matches at each neighbor and no anomalies are reported.

We show that if nodes are embedded in a 2-dimensional plane then 3 (and, in most all cases, 2) neighbors are sufficient for accurately validating spatial signatures. This implies that our protocol works in even relatively sparse networks. It also implies that in dense graphs it can work by designating only a small constant number of neighbors per node (as opposed to all neighbors) to realize the spatial signature validation primitive. Thus, in our protocol, authentic communications do not incur additional communication, whereas masqueraded communications can incur up to a small bounded number of communications. In our testbed experiments with the implemented protocol, this number is close to one.

Spatial-signature based message source authentication offers several benefits. First, a large amount of overhead incurred by cryptography operations and key management protocols is saved by the network. Second, it enables simple and efficient protocols for authentication, non-repudiation and integrity. Third, attacks created by compromised content-dependent signatures are not possible. Last but not least, it is resilient to node compromise and to node collusion. Conventionally, after more than a certain number of nodes are compromised, the security of the network is substantially decreased, whereas if the trust relationship is built only on spatial signatures, the damage caused by compromised nodes is regionally limited; likewise, collusion resistance can be achieved based on simple density arguments. To the best of our knowledge, we are the first to use the concept of spatial signature for authentication and related security properties.

The rest of this paper is organized as follows. In Section II, we pose the system model and the requirements that the spatial signature protocol must meet. We then describe the desired characteristics of the spatial signature primitive and, assuming the existence of the primitive, design a protocol based on that primitive for validating authenticity of the message sender, in Section III. In Section 4, we experimentally show how well RSSI and LQI suffice for realizing the spatial signature primitive. We discuss the realization of the primitive in Section 5 and its experimental evaluation in Section VI. Section VII reviews related work on authentication and related security properties in wireless sensor networks. We make concluding remarks and discuss future work in Section VIII.

II. System Model and Problem Statement

System Model. The system consists of a network of static, resource-constrained wireless nodes, which we refer to as motes, embedded in a K-dimensional space, where $K > 0$. Motes communicate with their peers in the network or with one or more base station nodes, in case the network has any base stations. Note that we do not require the presence of base stations.

We assume the following system properties:

- When motes are first deployed, there is an initial period during which no adversary is present, so motes can complete some computation towards trust establishment.
- Mote communications are all at the same power level. Their communication range is not however assumed to be isotropic. It follows that the sizes of the communication regions of different motes may be different.
- Each mote has a neighborhood degree of at least Δ, where Δ is a nonzero natural number constant which may be as small as three. Motes have access to a neighborhood service.
- Mote communications are atomic: a message sent by a mote is either received by all neighbors at approximately the same time or by none of its neighbors. In contrast, the time difference between any two communications from the mote is non-negligible.
- Mote communications are not directional.

Threat Model. The adversary in this wireless setting has, in the spirit of Dolev-Yao, the capability to:

- Eavesdrop on messages in its reception range, passively analyzing the method content and without revealing its presence to the motes.

- Block messages sent within its interference neighborhood.
- Replay older messages, possibly modifying the payload.
- Inject messages to motes, so as to disrupt the sensor application, to render the data flow corrupt or incomplete, or to attempt to compromise them and thus be able to launch further attacks. Injected messages can reach only the nodes in the neighborhood of the adversary.

The adversary may physically operate via devices other than the motes or it may operate via motes that it compromises. If a mote is compromised, its state becomes known to the adversary. Compromised motes may collude with other compromised motes within their communication range to launch attacks.

We assume that the adversary knows the spatial signature of each mote. We assume that the number of compromised neighbors of any mote do not exceed δ, where $0 \leq \delta < \Delta - K$. (By way of motivating this assumption, we note that, under this assumption, we intend for non compromised motes to enjoy their security properties despite these adversary capabilities. If this assumption is violated, we intend for the area of influence of a region of compromised motes to be strictly bounded.)

In terms of communication ability, unlike a non compromised mote, an adversary can communicate at multiple power levels, to attempt signal mimicry attacks. Like a non compromised mote, however, at each of these levels adversary messages reach Δ motes. Also, adversary communications are not directional.

(We will discuss later how to relax the system and the adversary communication assumptions.)

Notation. We let i, j, k, and l range over motes. We denote the adversary by A; when the adversary is at only one location, we ambiguously refer to that location by A as well.

Problem Statement. Our goal is to design a crypto-free protocol that robustly validates the authenticity of a message purporting to be sent from a mote, in the presence of the adversary described above. The protocol should suffice to achieve authentication, non-repudiation, as well as integrity, as follows. Consider the trivial protocol:

$$j \rightarrow k: \quad m$$

where j and k are motes and m is a message. If k receives message m, the primitive should suffice for k to (i) independently know that j (and no other node) is the sender, without knowing the content of m; (ii) explicitly prove that j is the sender of m, again without knowing the content of m; (iii) explicitly prove that j has received an earlier message m that k sent, knowing that m is the acknowledgement of a message m' that k sent earlier; and (iv) independently know that the message has not been modified, knowing that m is of the form $n, hash(n)$ but without knowing the content of n.

Note that property (i) implies authentication, properties (ii) and (iii) imply non-repudiation, and property (iv) implies integrity. By robustness of validation, we mean that the probability of the primitive failing is low if m is indeed sent by j and is high if m is sent by an adversary pretending to be j.

The problem then is to identify a primitive that robustly enables crypto-free communications and to use the primitive to design a message authenticity validation protocol with the following properties:

- *Lightweight.* Mote processing and communication overhead in realizing the primitive is low. By the same token, the latency introduced is low.
- *Scalability.* Mote processing and communication overhead in realizing the primitive scales efficiently as the density of or the number of motes grows.
- *Compatibility.* The primitive is easily incorporated into the network stack, i.e., it depends minimally (if at all) on particular network protocols. By the same token, it does not prohibit other security services (e.g. based on cryptography) from coexisting.
- *Compromise containment.* The impact of compromised motes in a region that violates the limit of δ, is contained to only their interference area.
- *Availability.* The primitive should not be vulnerable to denial of service attacks, even distributed ones.

III. Using the Spatial Signature Primitive

In this section, we describe the spatial signature concept, its desiderata, and (assuming that the desiderata can be realized) a protocol that uses

spatial signatures to authenticate the source of a message. We relegate the design of the primitive that realizes the desiderata to the next two section.

The Concept. As described in the introduction, the spatial signature of mote j is a physical characterization of the signal on the one-hop neighboring motes of j when j sends a message. The signature is evaluated by letting each neighbor sense the channel during j's message send to collect a predetermined number of samples.

Desiderata. For a small number of samples to suffice for the neighbors to reliably and accurately associate the evaluated characteristic with that of j, the following are the desiderata of the spatial signature:

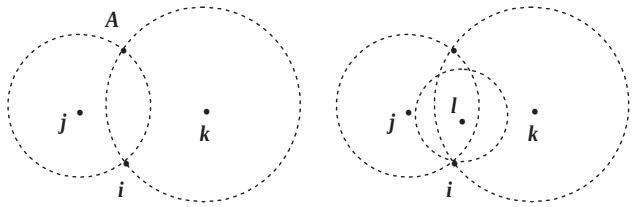
- *Stability:* The characteristic should be stable over time.
- *Uniqueness:* While the evaluated characteristic at a single neighbor need not be unique, the characteristic collectively determined by the one-hop neighbors should be unique.
- *Easy to learn:* Limited resource should suffice for the one-hop neighborhood of motes to determine and store the characteristic.

Neighborhood Size. Before we describe the protocol that uses spatial signatures to authenticate the source of a message, let us consider the minimum number of uncompromised motes in a neighborhood that are needed to achieve the uniqueness of spatial signatures.

Assume hypothetically that the physical characteristic enables perfect ranging at each receiver, i.e., it yields the distance from the transmitter to the receiver. In this case, we have

Proposition 3.1: If motes are in a K-dimension space, $K+1$ neighbors in general (i.e. non-degenerate) position suffice for defining a unique spatial signature for motes.

The geometric argument underlying this proposition is straightforward, and is in essence that of ranging-based mote positioning. We illustrate with an example in $2-D$ space. If a mote i has only one neighbor j, then an adversary A located anywhere on the circle of radius $dist_{ij}$ centered at j can convince j that A is i. If i has two neighbors, j and k, as shown in Figure 1(a), A can convince both j and k that it is i by being located at point A. But in Figure 1(b), where i has three non-collinear neighbors j, k and l, there exists no location at which A could be confused with i. By the same token, i cannot hide its identity from j, k, and l when it sends a message. Thus, 3 (which is $K+1$ in this example) neighbors in general position suffice for defining a unique spatial signature.



(a) Adversary at A cannot be distinguished from i by j and k (b) Adversary cannot imitate i without being detected by j, or k, or l

Fig. 1. Example of minimum density required for *uniqueness*.

In practice, accurate ranging via radio channel sensing is complicated by noise, calibration, and measurement variation issues, and is consequently unlikely to be lightweight enough for our purpose. However, if we can ensure stability of the physical characterization of the link, then the fact that the receiver signal is not characterized by a power loss captured by an idealized large-scale fading wireless model need not imply that the necessary neighborhood size must increase. On the contrary, we can

Claim 1: If motes are in a K-dimension space and ranging is imperfect, less than $K+1$ neighbors in general position suffice with high probability for defining a unique spatial signature for motes.

The claim is based on the fact that in our model an adversary at A cannot broadcast a message to all motes in the neighborhood of i while controlling the signal power level induced at each of the receivers independently to match that induced by i. (In fact, even if the model were weakened to allow it, the adversary cannot simulate a broadcast with a sequence of "directed" messages at multiple power levels to match the signature of i, if the neighborhood motes were time-synchronized well enough to distinguish between single and multiple transmissions). So the adversary is limited to finding a location at which sending a message will reproduce the exact "ranging-error-prone" spatial signature of i. In Figure 1(a), for example, suppose the ranging error yields an estimate of $dist_{ij}$ and

$dist_{ik}$ in the intervals $(dist_{ij} - e, dist_{ij} + e)$ and $(dist_{ik} - e, dist_{ik} + e)$ respectively, for some constant $e > 0$. If these errors are known to the adversary but are not correlated, then the adversary will be forced to search all points in the area of approximate size $2e \times 2e$ to determine whether there is any point at which it can reproduce the spatial signature of i. Thus, even with 2 neighbors in a plane the probability of violating uniqueness is low.

We will validate this claim in more detail in the next section, where we evaluate specific physical characteristics of radio signals.

A. Protocol for validating spatial signature

The basic idea of the protocol is *cooperative defense with anomaly detection*. If an adversary sends a message —even if at an alternative power level chosen to achieve signal strength mimicry— so as to impersonate mote j, an anomaly in the spatial signature of j will be detected by at least one of the following: (i) the intended receiver(s) of j, or (ii) some of the one-hop neighbors of j (in case the message is not a broadcast message) or (iii) a mote which is not a neighbor of j. A detecting mote will report the anomaly after a random delay unless it itself is the only intended recipient or it has already overheard another mote report an anomaly with this message.

At each intended receiver k, upon receiving a message purporting to be from j, k will measure the physical characteristic of the message and compares the measured value with the local signal signature it expects from j. If these values match, then k will wait for a short, pre-determined time and then accept the message, unless it receives a report of an anomaly with this message. If these values do not match, then it will not accept the message and it reports an anomaly after a random delay, unless t has already heard a corresponding report.

Example. Consider Figure 2, where the number of dimensions, K, is 2 and so, by model assumptions, $\Delta > 2$. Since all motes have at least Δ neighbors, it follows that all motes are within communication range of each other. Mote k thus knows the spatial signature induced by mote i at k, and so a message intended for j claiming to be from i but actually from A will be reported by k as an anomaly. Incidentally, if one of i or k were outside the communication range of A and none of i, j, and k were to detect an anomaly when A sends a message to j claiming to be from i, then there must exist some other mote in the network (not currently pictured) that would have reported the anomaly regardless of whether it were within the communication range of i or not. (End of example.)

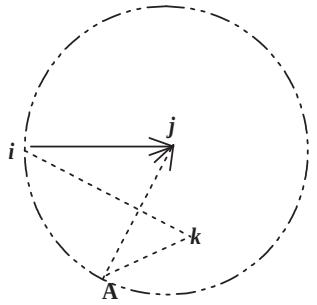


Fig. 2. The adversary A attempts to imitate mote i to mote j, but is detected by mote k.

The protocol deals with compromised motes as follows. If a compromised mote chooses not to report an anomaly, the assumption that the number of neighboring motes exceeds $\delta + K$ implies that at least one of the more than K non-compromised motes will report an anomaly.

Alternatively, if a compromised mote reports an anomaly even though its signature matches, then this may cause the receiver to falsely reject an authentic message. (Note that such a "false positive" report can be successful only if it comes from a neighbor of j, otherwise signature validation of the report itself will counteract the impersonation.) Our protocol does not attempt to remedy this sort of adversary attack, which may result in needless retransmissions, because the adversary is already in a position to simulate such an attack by frequency jamming when it detects the channel as being busy.

The pseudo code of the protocol is provided in Listing 1.

Listing 1. Message Source Authentication Protocol

```
bool check_validity(REPORT){
    if REPORT is a valid anomaly report
        return true;
    else
        return false;
} event Receive(MSG) { if MSG is of type data{
    if MSG comes from an unknown node
      or MSG->spatial_signature does not match{
        schedule an anomaly report;
        start anomaly_report_timer();
```

```
    }
    else{
          put MSG into a buffer;
          start accept_timer();
    }
  }
else if MSG is an anomaly report{
    if check_validity(MSG)
        count this report;
    else
        ignore;
  }
} event accept_timer_fired() {
 if no anomaly report is received
    accept that buffered MSG
else
    discard MSG;
 }
event anomaly_report_timer_fired() { if no similar anomaly report is
received
    send anomaly report;
else
    discard report;
}
```

This basic protocol idea can be refined for networks which have high density: in this case, not every neighbor of j has to maintain its spatial signature, but only a designated subset of Δ neighbors have to do so. The designated subset of motes would be chosen during the initial, secure training period of the network.

Recall that $\Delta > \delta + K$, so even when δ neighbors of j are compromised, there are more than K non-compromised motes neighboring j that will participate in the validation of any message purporting to be from j. By Proposition 3.1, this refinement of the protocol remains sound. (The same argument holds even when the chosen size of the designated subset is any constant number that is smaller than Δ but larger than $\delta + K$.)

B. Protocol Overhead & Collusion Resistance

Protocol Overhead. (1) If the adversary launches no attacks, the only cost is the local computation to validate signatures, which is negligible and considerably more efficient than traditional key based approaches where significant energy is spent in the absence of attacks. (2) If the adversary falsely sends anomaly reports, the only additional cost is the local computation to validate the anomaly reports, plus any resulting message retransmissions. (3) If the adversary impersonates another in sensing a message, this leads to only one anomaly report when all neighborhood detectors can overhear the report. Extra transmissions may occur when the report is not overheard by some of the other detectors.

The number of extra transmission is upper bounded by Δ, the constant number of designated detectors in every neighborhood, indicating the scalability of the protocol as network density increases. In fact, it is also upper bounded by min(Δ,5). To see this, consider Figure 2, where any mote placed in the disk neighborhood that contains i, j and A could detect the adversary A. If the communication model were a perfect unit disk, the worst case transmission overhead would occur when all reporters are on the disk boundary and just outside each others communication range. In this case, up to five messages could be sent to inform everyone in the disk. In practice though, the common neighborhood of i, j and A is likely to be smaller than this unit disk and, as we have argued before, with high probability a degree of 2 suffices for a 2-D network, so the number of transmissions would be fewer than min(Δ,5).

Collusion Resistance. When compromised motes in a neighborhood collude by being silent, as long as there exist at least $K+1$ non-compromised motes in the neighborhood, the anomaly will be detected and reported; even fewer non-compromised motes may suffice, as discussed before. When the number of colluders increases to dominate some region, i.e. the region has less than $K + 1$ non-compromised motes witness messages per sender mote, the impact of collusion is contained to the region's interference range by virtue of the locality of the protocol.

IV. RSSI & LQI as a Basis for Spatial Signatures

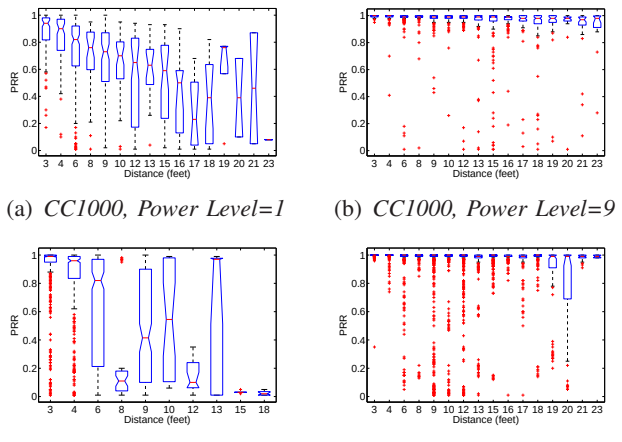
In this section, we discuss two physical features that can suffice to realize the spatial signature primitive and experimentally evaluate them in terms of the spatial signature desiderata of stability, uniqueness, and ease of learning. How the primitive itself is realized using them is the topic of the next section.

Specifically, the two features are the Received Signal Strength Indicator (RSSI) and the Link Quality Indicator (LQI) respectively. RSSI measures the received radio signal strength (i.e., the energy

integral, and not the quality of the link). RSSI output is often a DC analog level, which can be sampled by an internal ADC. RSSI is available in many mote radios such as the *CC1000, TR1000,* and *CC2420*. In motes with the *CC2420*, for example, the RSSI value is averaged from over 8 symbol periods (128 μs);

LQI measures the strength and/or quality of a received message [13]. LQI reflects not just a physical property of a link, but also the temporal variation along the path and hardware calibration. It is supported by *802.15.4* compatible radios such as *CC2420* [13] but not on other commonly used motes.

Previous work has indicated that RSSI is not a good indicator of link quality in older radios such as $CC1000$ and $TR1000$ [18], [19], but it performs better in newer radios such as the $CC2420$ [14]. While previous work has focused on the correlation of these indicators with link quality, we do not. In contrast, we study below the properties of stability, spatial uniqueness, and ease of learning signatures based on these indicators. In particular, we find that metrics derived from the RSSI (alternatively, from LQI) values over one or more message transmissions on a link are statistically stable for that link, especially for the *CC2420* motes. We also find that these features provide reasonable uniqueness that can be used to differentiate a mote from another. Therefore we apply either of these two physical features in realizing the spatial signatures primitive.



(a) *CC1000, Power Level=1* (b) *CC1000, Power Level=9*

(c) *CC2420, Power Level=2* (d) *CC2420, Power Level=3*

Fig. 3. Density of the target testbed at different power level.

A. Experimental evaluation of RSSI and LQI

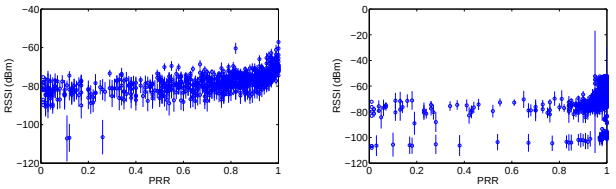
We begin by describing the testbed used in our experiments. All of our experiments involve 42 motes with the $CC1000$ radio and 42 motes with the $CC2420$ radio; both networks are organized in a rectangular grid of 6 x 7 motes. (The identity of the testbed is anonymized for blind review.) Each $CC1000$ mote consists of a $4MHz$ ATmega128L microcontroller, $128KB$ of flash and $4KB$ of RAM; the radio frequency is $433MHz$. Each $CC2420$ mote consists of a $8MHz$ MSP430 microcontroller, and operates at $2.4GHz$. It has $48KB$ of flash and $10KB$ of RAM. We use TinyOS in our experiment on both types of motes [1].

The first dataset we collected is based on multiple runs: in each run a different power level is chosen and each mote is given a turn in a round-robin manner to transmit 700 broadcast messages (one per 128 millisecond). We measure the RSSI, the LQI, and the Packet Reception Rate (PRR) between each pair of motes.

This dataset helps us select networks of different density/neighborhood sizes, by choosing an appropriate power level. As shown in Figure 3, the testbed is roughly a one-hop network for the $CC1000$ and $CC2420$ motes at power level 9 and 3 respectively. At these power levels, almost every mote is able to reliably communicate with any other one. At the lowest power level 1 for the $CC1000$ motes, and power level 2 for the $CC2420$ motes, the testbed becomes a 3-5 hops network, where a mote can only reliably talk to some neighboring mote. (Of course, the Figure reconfirms the well known facts that distance does not always has a negative impact on PRR, that radio connectivity is irregular and non-isomorphic [18], and the complex relation between PRR and RSSI/LQI.)

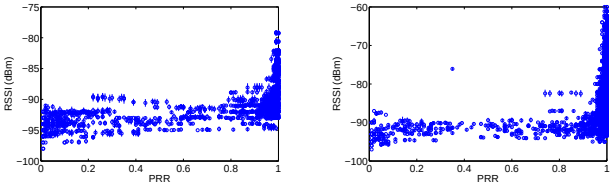
B. Evaluating Stability

We study here the variability of RSSI and LQI on any given link. Figures 4 and 5 plot RSSI versus PRR collected from a window of 100 messages. Each link is represented by a line centered at its mean RSSI value. Half the length of each line measures the standard deviation of the RSSI value on that corresponding link. ¿From the plots, we see that while RSSI measurements on a given *CC1000* mote link could have about 10dBm deviation, the



(a) *Power Level=1* (b) *Power Level=9*

Fig. 4. Variability of RSSI on the *CC1000* motes.

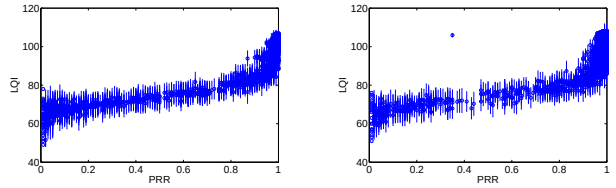


(a) *Power Level=2* (b) *Power Level=3*

Fig. 5. Variability of RSSI on the *CC2420* motes.

measurements on a given *CC2420* mote link experience very small variance (less than 2dBm). This suggests that a primitive derived from multiple RSSI measurements per link can be quite stable for *CC2420*, confirming the results recently reported in [14].

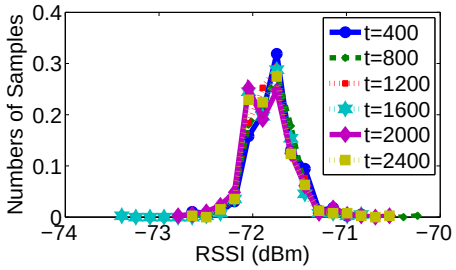
Figure 6 shows the variability of LQI per link. We find that it has a larger range of variation than RSSI at both power levels. (On a side note, we see that its mean value over many messages seems to have a better correlation with PRR, for all qualities of links.) Next, we consider in more detail the short-



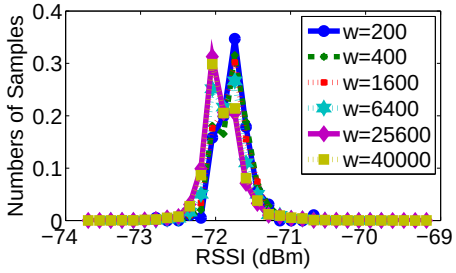
(a) *Power Level=2* (b) *Power Level=3*

Fig. 6. Variability of LQI on the *CC2420* motes.

term as well as long-term RSSI/LQI variation on individual links. We carried out another series of experiments in which one corner mote sends $40,000$ messages in total (one per 128ms) at power level 9 for the *CC1000* motes and power level 3 for the *CC2420* motes. We calculated the histogram over a given window interval $(t, t+w)$, where t is the start time and w is the window size (number of messages). The approximated normalized distribution of RSSI/LQI on a given link can be characterized given t and w. Since different links are observed to have similar stability property, we plot two cases for a representative link: (1) variable w and fixed t; (2) variable t and fixed w In Figures 7, 8 and 9. We see



(a) *RSSI, fixed w=400 packet time*



(b) *RSSI, fixed t=200 packet time*

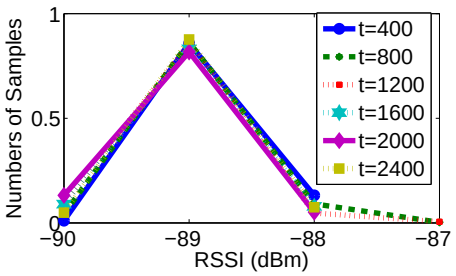
Fig. 7. RSSI distribution on the *CC1000* motes.

that across different start times as well as different window periods, RSSI ranges over roughly the same distribution. These observations imply that RSSI does not change dramatically over time. Instead, it is statistically stable. Moreover, if we compare the number of discrete RSSI value a link takes in Figures 7 and 8, we see that only a few values (four in this example) are assumed on the *CC2420* motes, while tens of values appeared on the *CC1000* motes.

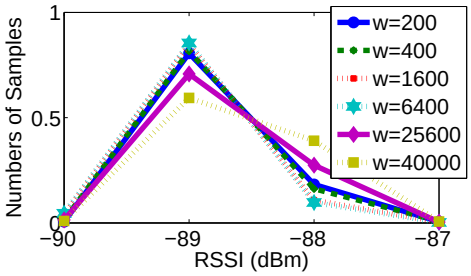
The findings for LQI on the *CC2420* motes, as shown in Figures 9(a) and 9(b), are likewise. This motivates us to model the RSSI/LQI value in a statistical manner. Since the frequency of each sampled RSSI/LQI value at different start time t and window size w is relatively consistent, we may model the RSSI on the *CC1000* motes and LQI on the *CC2420* motes with a mixture of Gaussians or, for simplicity, a one-degree Gaussian function:

$$p(x|\Theta) = \sum_{j=1}^{M} \alpha_j p_j(x|\theta_j) \qquad (1)$$

where the parameters are $\Theta = (\alpha_1, .., \alpha_M, \theta_1, ..., \theta_M)$ such that $\sum_{j=1}^{M} \alpha_j = 1$, $\theta_j = \{\mu_j, \sigma_j\}$ and each p_j is a normal density function $n(\mu_i, \sigma_i)$ as in Equation 2. M is the

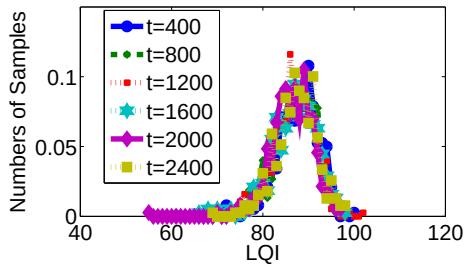


(a) *RSSI, fixed w=400 packet time*

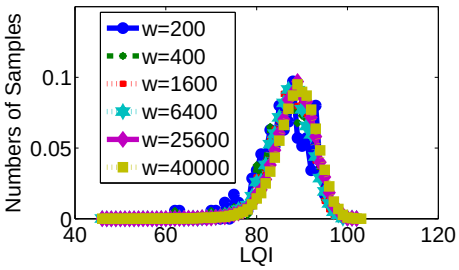


(b) *RSSI, fixed t=200 packet time*

Fig. 8. RSSI distribution on the *CC2420* motes.



(a) *LQI, fixed w=400*



(b) *LQI, fixed t=200*

Fig. 9. LQI distribution on the *CC2420* motes.

number of probability density components. When $M = 1$, $p(x|\Theta)$ is simply a normal distribution.

$$p_j(x|\theta_j) = \frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{(x-\mu_j)2}{2\sigma_j 2}} \quad (2)$$

For the *CC2420* motes, since the RSSI assumes only a small number of values over a long time, a simple histogram suffices to reflect its distribution:

$$p(x_i|\Theta) = \sum_{j=1}^{M} I(x_i = v_j) \times p(x = v_j) \quad (3)$$

where $p(x = v_j)$ is the prior probability for the j^{th} instance in the histogram table, M is the number of instances, and I is an indicator function: $I(x_i = v_j) = 1$ iff $x_i == v_j$, and $I(x_i = v_j) = 0$ otherwise.

Interestingly, if we compare the results in Figures 7 and 8, we see that the RSSI measurement has become more precise. We envision that such measurement will be even more precise in future devices, in which case the performance of the proposed primitive will become even more promising.

C. Evaluating Spatial Uniqueness

In this subsection, we experimentally quantify the dissimilarity between RSSI/LQI measures induced by different motes at their respective neighbors. Specifically, we model dissimilarity as follows:

$$\mathcal{D}_{o_1,o_2} = \frac{1}{||\mathcal{N}_{o_1,o_2}||} \sum_{q \in \mathcal{N}_{o_1,o_2}} \phi_{o_1,o_2,q} \quad (4)$$

where o_1 and o_2 are two motes, and $\mathcal{N}_{o_1,o_2}$ is the neighborhood view of o_1 and o_2. ϕ is a function that measures the difference between two distributions,

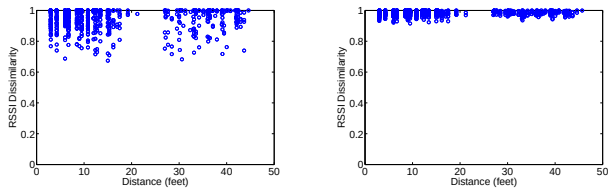
$$\phi_{o_1,o_2,q} = \frac{1}{2} \int_{-\infty}^{\infty} |p(x|\Theta_{o_1,q}) - p(x|\Theta_{o_2,q})| dx \quad (5)$$

Here $p(x|\Theta_{o,q})$ refers to the probability for variable x given the distribution of a feature on the $link(o, q)$. For example, if one feature on $link(A, B)$ has a discrete distribution: $p(x = 0) = 0.6$, $p(x = 1) = 0.4$; and the same type of feature on $link(A, C)$ has a discrete distribution: $p(x = 1) = 0.3$, $p(x = 2) = 0.7$; then $\phi_{B,C,A}$ will be $(|0.6 - 0| + |0.4 - 0.3| + |0 - 0.7|)/2 = 0.7$. For the normal distribution case, ϕ is then the sum of the area that two normal density functions do not share. It is not hard to see that if the two distributions are close, then ϕ tends to 0; otherwise, it tends to 1.

Thus, $\mathcal{D}_{o_1,o_2}$ in Equation 4 measures the average dissimilarity of o_1 and o_2 among the neighborhood. If this value is sufficiently large, then we can use some threshold mechanism to distinguish one mote from another.

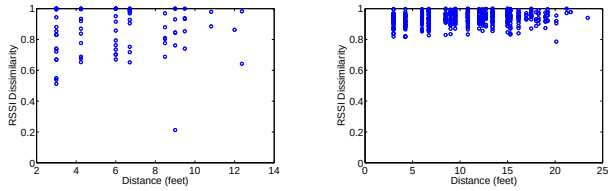
We use the round robin traffic dataset to estimate the dissimilarity. Each link takes the first 100 messages to establish the distribution. We apply discrete distribution for the RSSI on the *CC2420* motes, and a single normal density for the RSSI on the *CC1000* motes and LQI on the *CC2420* motes. The results

of dissimilarity of two motes at a certain distance are shown in Figure 10, 11, and 12. Note that each point is the average dissimilarity to the whole common neighborhood of a pair of motes. We see



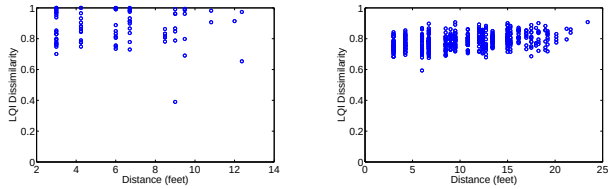
(a) *CC1000, Power Level=1* (b) *CC1000, Power Level=9*

Fig. 10. Dissimilarity of RSSI vs. Distance on the *CC1000* motes.



(a) *CC2420, Power Level=2* (b) *CC2420, Power Level=3*

Fig. 11. Dissimilarity of RSSI vs. Distance on the *CC2420* motes.



(a) *CC2420, Power Level=2* (b) *CC2420, Power Level=3*

Fig. 12. Dissimilarity of LQI vs. Distance on the *CC2420* motes.

that RSSI on the *CC1000* motes has relatively large dissimilarity at both the lowest power level and a relatively high power level, while RSSI and LQI on the *CC2420* motes have high dissimilarity at the higher power level, and a few motes have low dissimilarity values at the lower power level. The reason could be that at the lower power level, the range of values that RSSI/LQI actually takes might be relative small, so that a few pairs of motes have relatively low dissimilarity. However, we see that all of them are still above some reasonable threshold values.

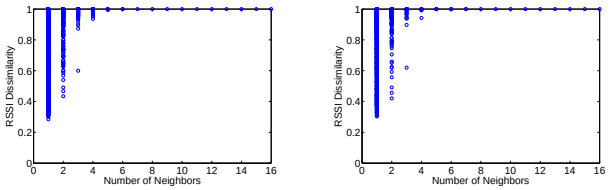
We note that for distinguishing between different source motes the maximal value of dissimilarity is more relevant than the average value, since an anomaly does not need to be reported by every mote in the neighborhood. As long as there is a non-compromised mote that reports, the neighborhood will be informed about the anomaly. However, we present the average dissimilarity rather than the maximum value to more conservatively support the arguments for the spatial uniqueness desideratum.

For scalability, we are interested in quantifying the dissimilarity of two motes relative to a constant number of neighbors. To avoid visually lowering the effectiveness of a potential 'best' detector by averaging the dissimilarity values, especially when the neighbor size is small, we use the maximum value of the dissimilarity in the selected neighborhood because this value is the critical one to detect an anomaly. We plot the dissimilarity over the constant number of neighbors in Figures 13, 14, and 15. It is not surprising to see that there are many cases where two motes look similar to a single neighbor. However, when the neighbor size grows, the dissimilarity for any two motes increases. One interesting observation is that even in the case where only two neighbors exist, all the dissimilarity values are still above some reasonable threshold on both the platforms at different power levels, which somehow verifies our previous argument that two neighbors rather than three are actually very likely to be able to identify the difference between a legal mote and an impersonator. In the *CC1000* case, we see that any two motes have dissimilarity value 1 when the neighbor size is greater than four. A similar observation holds for RSSI in the *CC2420* case. Unlike the RSSI case, the dissimilarity with LQI grows slowly as the number of neighbors increases.

Figure 14 shows cases where a single neighbor is insufficient to detect an anomaly with RSSI. Notably, with LQI, as in Figure 15, there is no case with dissimilarity close to 0 with only a single neighbor. This strengthens the claim that an authentic mote may be distinguishable from an adversary. Overall, these results verify the claim about minimum density required to detect anomaly.

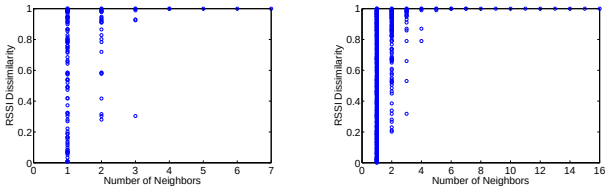
Next, we address the question of how easy is it for the adversary to use power control to simulate a RSSI/LQI signature at neighbors of j. To this end, we characterize the variability relationship between RSSI/LQI and distance. Ideally, RSSI would fade inversely as some power of the distance, but this is rarely exact in practice.

Figure 16 shows the correlation between RSSI



(a) *CC1000, Power Level=1* (b) *CC1000, Power Level=9*

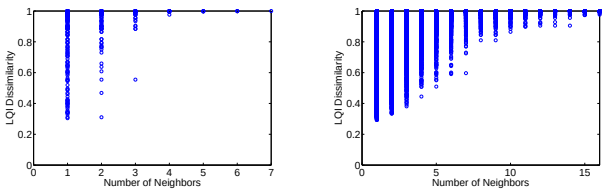
Fig. 13. Dissimilarity of RSSI vs. #Neighbors on the *CC1000* motes.



(a) *CC2420, Power Level=2* (b) *CC2420, Power Level=3*

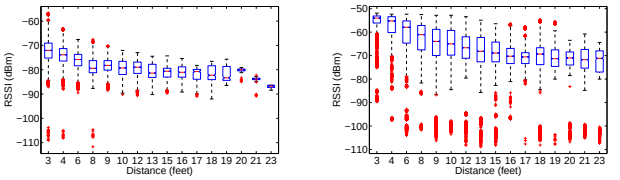
Fig. 14. Dissimilarity of RSSI vs. #Neighbors on the *CC2420* motes.

and distance on different platforms. We see that roughly RSSI falls off with distance but not uniformly though. For example, the mean RSSI value at distance 12, in Figure 16(c), is clearly larger than that at other distances (except 3 feet). Such an irregularity makes certain attacks even harder when they attempt to imitate the source by estimating the distance to the destination and applying corresponding transmission power. Although there are quite a few outliers in the *CC1000* motes, shown in Figure 16(a) and 16(b), it can be observed that far fewer outliers occurred in the *CC2420* platform, which suggests that this radio achieves better stability at the same distance. Figure 17 shows the relationship between LQI and distance on the *CC2420* motes. Since LQI implies the quality of the received messages, it is likely to have a higher value with shorter distance. However, LQI seems to be less correlated with distance than RSSI. For example, at power level 2, the mean LQI at distance 10 (in Figure 17(a)) is relatively high while the RSSI value at that distance (in Figure 16(c))

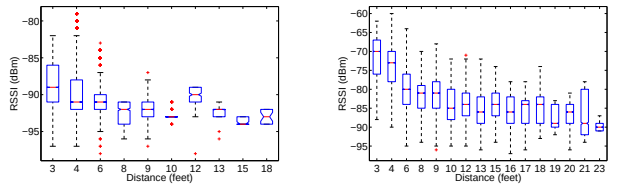


(a) *CC2420, Power Level=2* (b) *CC2420, Power Level=3*

Fig. 15. Dissimilarity of LQI vs. #Neighbors on the *CC2420* motes.



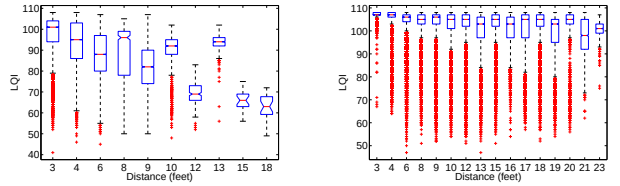
(a) *CC1000, Power Level=1* (b) *CC1000, Power Level=9*



(c) *CC2420, Power Level=2* (d) *CC2420, Power Level=3*

Fig. 16. RSSI vs. Distance.

is relatively lower compared with RSSI at other distance. At power level 3, it is easy to observe a virtual decreasing curve of RSSI over distance in Figure 16(d), while the same pattern does not occur on the LQI feature in Figure 17(a). The reason might be that LQI also indicates something more about temporal variation in addition to inherent quality of a link. Figures 16 and 17 suggest that neighbors at



(a) *CC2420, Power Level=2* (b) *CC2420, Power Level=3*

Fig. 17. LQI vs. Distance on the *CC2420* motes.

different distances have different observations of the features. And even at the same distance, the features may vary. Such an irregularity poses a challenge for an adversary to use power control at certain distance to simulate RSSI/LQI without being detected at first place. As argued before, it is even more difficult to choose a power level to simulate RSSI/LQI for multiple neighbors simultaneously.

D. Evaluating Ease of Learning

For efficiently realizing the primitive, we need a succinct way of accurately modeling the RSSI/LQI spatial signature. Let us assume that data samples are independent and identically distributed with distribution p. Therefore, the resulting density for

the sample is:

$$p(\mathcal{X}|\Theta) = \Pi_{i=1}^{N} p(x_i|\Theta) = \mathcal{L}(\Theta|\mathcal{X}) \quad (6)$$

Here N is the number of samples, and function $\mathcal{L}(\Theta|\mathcal{X})$ is the likelihood of the parameters given the data. In order to achieve the maximum posterior probability, we wish to find the Θ that maximizes $\mathcal{L}$ as follows,

$$\Theta^* = argmax_{\Theta}\mathcal{L}(\Theta|\mathcal{X}) \quad (7)$$

To achieve the maximum likelihood, we discuss two cases respectively:

Discrete Case: If the distribution p is discrete, it can be simply represented by a table of normalized sample frequency,

$$p(x = v_j) = \frac{\sum_{i=1}^{N} I(x_i = v_j)}{N} \quad (8)$$

where N is the number of samples. I is an indicator function: $I(x_i = v_j) = 1$ iff $x_i == v_j$, and $I(x_i = v_j) = 0$ otherwise, and $j = 1, \cdots, M$, where M is the number of instances.

$p(x_i|\Theta)$ can be simply obtained by

$$p(x_i|\Theta) = \sum_{j=1}^{M} I(x_i = v_j) \times p(x = v_j) \quad (9)$$

Gaussian Case: If the distribution p is a probabilistic model as follows,

$$p(x|\Theta) = \sum_{j=1}^{M} \alpha_j p_j(x|\theta_j) \quad (10)$$

where the parameters are $\Theta = (\alpha_1, .., \alpha_M, \theta_1, ..., \theta_M)$ such that $\sum_{j=1}^{M} \alpha_j = 1$, $\theta_j = \{\mu_j, \sigma_j\}$ and each p_j is a normal density function $n(\mu_i, \sigma_i)$. M is the number of density components. When $M = 1$, $p(x|\Theta)$ is simply a normal distributed function.

The likelihood function can be maximized using the EM (Expectation-Maximization) algorithm [2] iteratively as follows,

$$\alpha_j^{new} = \frac{1}{N}\sum_{i=1}^{N} p(j|x_i, \Theta^g) \quad (11)$$

$$\mu_j^{new} = \frac{\sum^{N^{i=1}} x_i p(j|x_i, \Theta^g)}{\sum_{i=1}^{N} p(j|x_i, \Theta^g)} \quad (12)$$

$$\sigma_j^{new} = \frac{\sum_{i=1}^{N} p(j|x_i, \Theta^g)(x_i - \mu_j^{new})(x_i - \mu_j^{new})^T)}{\sum_{i=1}^{N} p(j|x_i, \Theta^g)} \quad (13)$$

As described above, mathematically, the spatial signature of a mote is Θ, which is a set of parameters for a tabular distribution in the discrete case and a linear combination of Gaussian functions in the Gaussian case. It is easy to learn, without introducing too much computational overhead.

V. Realization of the Spatial Signature Primitive

In this section, we use RSSI/LQI measurements associated with messages and the model developed above to realize the spatial signature primitive.

The realization consists of two phases: the training phase and the verification phase. The former learns the model-parameter values for each node j, comprising one set for each neighbor of j that is designated to authenticate communications from j. The latter performs the message source authentication, by letting each neighbor collect one or more RSSI/LQI samples during receipt of one or more messages purporting to be from j, and then determining whether the likelihood of these samples being consistent with the model-parameter values exceeds some threshold; if not, the primitive announces the failure of the match.

Training Phase: The goal of the training phase is to learn the spatial signatures from selected neighboring nodes. As stated in Section II, we assume this phase is free from adversaries. In order to learn the spatial signatures, a mote actively/passively collects the RSSI and LQI (if available) samples from each of its neighbors. Since many networks apply beacons or control messages either for routing purpose or other specific services immediately after deployment, our protocol can take advantage of those traffic and build its initial statistical model in the training phase, thus no extra communication overhead is required to establish the trust among legal parties. If there is no such traffic that can be exploited, the protocol itself should generate some traffic for the training purposes. The training phase may consume some memory to buffer samples. However, all the buffers will be released after the models are obtained. The only memory consumed afterwards is to retain the model parameters, which is a small requirement.

For scalability, this service does not need to include all motes from which one can possibly receive a message. We may only keep the spatial signatures for a fixed number of neighbors based on the theory in Section III. A consequent question is whose spatial signature should be chosen to be maintained.

Of course, no mote should devote a significant amount of memory to maintain spatial signatures for a large number of neighbors. This problem may be dealt with in the same way as it is in other existing services (e.g. routing protocols) where there is a concept of neighborhood. This neighborhood service may exploit existing neighborhood information (if available) from other components. One possible way is to keep a fixed number of good neighbors with high quality link only. By this, the training cost is fixed. It does not suffer from the scalability problem. Of course, there is a tradeoff between the memory consumption for spatial signatures and resiliency to node compromise.

Verification Phase: In this phase, a mote verifies messages coming from others including both legal motes and adversaries by matching the primitive. To authenticate the incoming message(s), we calculate the similarity of observed sample(s) to a primitive corresponding to the claimed identity. In order to count two types of features with different type of distributions, we use the following weighted average to get the likelihood value:

$$\mathcal{L}(\Lambda_k^*|\mathcal{X},\mathcal{Y}) = \beta \times ln\mathcal{L}(\Theta_{k,1}^*|\mathcal{X}) + (1-\beta) \times ln\mathcal{L}(\Theta_{k,2}^*|\mathcal{Y}) \quad (14)$$

Here $\mathcal{X}$ and $\mathcal{Y}$ are two types of features, corresponding to two different distributions Θ_1 and Θ_2. $\beta, 0 \leq \beta \leq 1$ is a weighted coefficient to balance the significance of different features; log function is used to avoid numerical overflow; k is the index of the model. If only one type of feature is used (e.g. on the *CC1000* radio components), β is adjusted to be 0 (or 1).

The weighted likelihood obtained by Equation 14 is then compared with a predefined threshold. A message claiming from k is considered a *match* if and only if the following is true,

$$\mathcal{L}(\Lambda_k^*|\mathcal{X},\mathcal{Y}) > T_k \quad (15)$$

where Λ_k^* represents the trained model(s) for mote k, $\mathcal{X}$ and $\mathcal{Y}$ are testing features. T_k is the predefined threshold for mote k. A **false negative** occurs when a mote purporting someone else is falsely accepted; and a **false positive** occurs when a message from an authentic mote is falsely rejected. Obviously, threshold T is a key value in the decision process. The practical setting of T depends on the security requirement of the applications, considering the consequence of the occurrence of false positives and false negatives. In general, a small T is likely to accept more messages possibly including those from adversaries. Therefore, it may produce more false negatives. On the other hand, a large T has more chance to prevent false negatives, but may reject messages that are actually from legal motes, and produce more false positives. In the later evaluation, T is adapted during the training phase such that no false negative occurs. Of course it does not guarantee that no false negative happen afterwards, but it is very likely that false negatives would be quite low. The reason to have a relatively small T is that the damage created by false negatives is likely to be more severe than that created by false positives. The occurrence of false positives can be remedied by retransmissions. If the original source is under suspect, a receiver can ask for data retransmissions from other trusted neighbors.

Note that our scheme works for both packet level validation and batch level validation. The latter mode is especially useful for stream data applications such as network reprogramming [5], [15], [16]. It yields lower overhead, while providing more confidence to accept/reject the messages. Of course, some integrity mechanisms (e.g. hash functions) may be involved to ensure batch integrity.

VI. Evaluation of the Spatial Signature Primitive

In this section, we evaluate the primitive for both granularities. Specifically, we quantify the performance of a TinyOS implementation of the primitive, via testbed based experiments on the same 42 motes network and power levels, but with different traffic patterns.

To enable packet level validation, we modified the radio component of the *CC1000* motes to return a sequence of RSSI values (10 in this evaluation) during the receipt of a single message (the implementation returns only one RSSI value per packet). We note that in the batch level validation on the *CC1000* motes, our experiments use the default implementation, which produces only one RSSI value per packet; this is to study the impact of test sample size. For the *CC2420* motes, we are still in the process of changing the radio component code. However, packet level validation is still possible since RSSI on the *CC2420* is modeled as a discrete distribution, in current implementation of the primitive.

Experiment Design: Traffic Patterns. To study

the performance with different traffic patterns at different power levels, we carried out the following experiments:

1) Round robin without intentional interference. Each mote, in a round robin fashion, broadcasts a sequence of messages (one per 128 ms) with the first 100 messages being used for training purpose, and the rest of the messages for testing. No intentional interference is introduced. Every L (L is instatiated with 10, 5, and 2) messages are considered as a batch unit for testing. In each experiment, each mote not only claims its true identity, but also attempts to imitate any other one in the network. Therefore in each experiment, there are 42 true identities, and $42 \times 41 = 1722$ fake identities.
2) Round robin with intentional interference. The above experiment is repeated but with interference introduced, with two motes in the center of the network each periodically sending at a high frequency an interfering message preceded by a random delay (on average, one message is sent per 64ms). Note that these two special motes are also involved in the evaluation. They act as themselves, and mimic others as well. Notice that here intentional interference is incurred from inside of the testbed. We should point out that the testbed is indoor and coexists with several concurrent WiFi networks. Since 802.11 shares spectrum with the 802.15.4 radios, the resulting interference can also affect the evaluation on the *CC2420* motes.
3) Synthesized burst periodic traffic. Every mote randomly broadcasts a message (1000 in total) on its own behalf, but also in impersonation of any other mote in the network. The frequency of sending on each mote is one per 3 seconds on average. Again, the first 100 messages are used for training, the rest for testing.

Batch level validation is evaluated in the first two experiments that mimic simple stream data applications, while packet level validation is performed in the last experiment which simulates certain data collection applications.

Efficiency Microbenchmarks. Table I compares the memory requirements and time efficiency of our primitive with that of digital signatures, as obtained from a recent publication [6], for *CC2420* motes.

Regarding memory, the *RSA-1024* RAM requirement is calculated for authenticating one base station only. If each mote is designed to authenticate multiple parties, the $RAM/party$ in [6] could be reduced by applying memory optimization, but would still be (a) rather difficult to realize for dense networks because of the large RAM requirement and (b) substantially higher in $RAM/party$ compared to our 20 bytes/party.

Incidentally, for the *CC1000* motes, we estimate the same RAM requirement and approximately double the computation time, as the latter's microcontroller operates half as fast. And we note that implementing the training and testing phases requires, in addition to normal integer operations, math functions such as exponential function and log function. Since our motes only support integer operations, we use the Taylor series to approximate exponential and log functions.

Regarding time, the speed of our primitive for a single (local) validation operation is about 5000 times faster than the RSA-based approach. RSA validation is deterministic, whereas ours is statistical and can involve false positives, which can imply additional retransmissions to finish data communication. In this sense, the average time requirement for our primitive would be a little more than that of a single local validation; that said, we will show next that false positives are rather rare.

Method	Time	RAM/party
RSA-1024	0.7 s	529 bytes
Our Primitive	0.13×10^{-3} s	$\approx$ 20 bytes

TABLE I

Time for a single testing operation and RAM requirements in an implementation of the primitive for *CC2420* motes. Data for RSA approach is from [6]. Primitive computation time in our protocol over 20000 testings.

Parameter Setting and Network Density. For the *CC1000* motes, we use a one degree Gaussian function to model the RSSI distribution. For the *CC2420* motes, we use a histogram of size of 12 for the RSSI feature, and a one degree Gaussian function for the LQI value. The coefficient β is set to be 0.8 to give more weight to the RSSI feature because of its better stability. The network density is

type of motes	Min. D.	Max. D.	Ave. D.
CC1000, PL=1	3	19	8.7
CC1000, PL=9	9	37	31.5
CC2420, PL=2	3	8	4.2
CC2420, PL=3	36	41	39.1

TABLE II

Network Density. PL=Power Level. D.=Degree.

approximately reflected by the size of the motes in the neighborhood. Table II lists the minimum, maximum, and average degree of the target networks composed by different platforms. We see that the average degree for the *CC1000* motes at the lowest power level 1 is about 9, while the average degree for the *CC2420* motes at power level 2 is only about 4. The networks at these two lower power levels are relatively sparse. On the other hand, the degree of the network for the *CC1000* motes at power level 9, and *CC2420* motes at power level 3 is relatively high, which indicates that the network at these power levels is relatively dense. The results of density in Table II are consistent with the boxplots in Figure 3.

Batch Level Testing Results. We evaluate the primitive for different batch sizes (10, 5, and 2) separately. In the case of 5 messages a batch, for example, each mote repeats 120 times claiming its true identity, and 120 times claiming a false identity corresponding to every other victim mote. Since there are 42 motes in total (for each platform), the maximum total number of verification cases is $42 \times 120 = 5040$, and the maximum total number of imitation cases is $42 \times 41 \times 120 = 206640$. Note that the network may not be completely connected, especially at the lower power level, so no verification or imitation needs to be performed between motes who can not talk to each other.

Tables III and IV list the validation results for the 10 messages per unit case on the *CC1000* motes and *CC2420* motes respectively. We see that there are a few false positives for the *CC1000* motes at the lowest power level, 0.12% for the no-interference case and a little more, 0.62%, for the with-interference case. What is encouraging is that no false negative or false positive occur at the higher power level. The reason might be that the signal strength seems to be more stable at the higher transmission power level, as shown in Figure 4. Another promising result is that no error happens for the *CC2420* motes, even at the lower power level where there are many lossy links. This is mainly due to the increased stability of the feature, as mentioned above.

If we decrease the size of a testing unit, a mote is likely to have less confidence to verify/reject a batch. The results for the 5 messages a batch, and 2 messages a batch are listed in Table V, Table VI, Table VII, and Table VIII. We see that false positives have been increased on both platforms at different power level, but still no false negative occurs. Particularly, most false positives are less than 3% except in the with-interference case on the *CC1000* motes at power level 1 when the batch size is 2, where the false positive reaches 5.68%.

Overall, false positives may increase as the batch size decreases, but they are all below 6%. The results validate the robustness of the proposed primitive at batch level, using the default radio implementation.

CC1000 motes	False Positive	False Negative
N.I., PL=1	0.12%	0%
N.I., PL=9	0%	0%
N.I., PL=1	0.62%	0%
W.I., PL=9	0%	0%

TABLE III

Testing results for the *CC1000* motes with **10 messages per testing unit**. N.I.=No Interference; W.I.=With Interference; and PL=Power Level.

CC2420 motes	False Positive	False Negative
N.I., PL=2	0%	0%
N.I., PL=3	0%	0%
W.I., PL=2	0%	0%
W.I., PL=3	0%	0%

TABLE IV

Testing results for the *CC2420* motes with **10 messages per testing unit**. N.I.=No Interference; W.I.=With Interference; and PL=Power Level.

Packet Level Testing Results. Tables IX and X show the results of the evaluation at the single packet level. False positives on the *CC1000* motes are 0.94% and 0.33% at power level 1 and 9 respectively. They are even less than that at batch level of unit size 2 in Table VII. The reason is that we applied default radio implementation where only

CC1000 motes	False Positive	False Negative
N.I., PL=1	0.69%	0%
N.I., PL=9	0%	0%
W.I., PL=1	2.66%	0%
W.I., PL=9	0%	0%

TABLE V

Testing results for the *CC1000* motes with **5 messages per testing unit**. N.I.=No Interference; W.I.=With Interference; and PL=Power Level.

CC2420 motes	False Positive	False Negative
N.I., PL=2	0%	0%
N.I., PL=3	0%	0%
W.I., PL=2	0%	0%
W.I., PL=3	0%	0%

TABLE VI

Testing results for the *CC2420* motes with **5 messages per testing unit**. N.I.=No Interference; W.I.=With Interference; and PL=Power Level.

CC1000 motes	False Positive	False Negative
N.I., PL=1	1.25%	0%
N.I., PL=9	0%	0%
W.I., PL=1	5.68%	0%
W.I., PL=9	0.018%	0%

TABLE VII

Testing results for the *CC1000* motes with **2 messages per testing unit**. N.I.=No Interference; W.I.=With Interference; and PL=Power Level.

CC2420 motes	False Positive	False Negative
N.I., PL=2	0%	0%
N.I., PL=3	0.03%	0%
W.I., PL=2	0.09%	0%
W.I., PL=3	0.02%	0%

TABLE VIII

Testing results for the *CC2420* motes with **2 messages per testing unit**. N.I.=No Interference; W.I.=With Interference; and PL=Power Level.

one RSSI value is produced per packet for the batch level validation. In the case of testing unit size 2, only 2 RSSI values are used for testing. However, we used modified implementation where 10 RSSI values are produced per received message for the packet level validation, so that a mote has more confidence to accept/reject a message. In addition, if we compare this result with that of batch size 10, we find that it is somewhat worse although the number of RSSI values for a single testing is the same. The reason is probably that the traffic used in the packet level validation is worse than that in the batch level validation. As described above, every mote periodically sends a message with high frequency, so there could be more interference occurred in the network. The results for the *CC2420* motes are promising too, 0.053% and 1.05% false positives at power level 2 and 3 respectively.

Overall, this result shows a high level of accuracy of the proposed primitive at per packet level. It validates the robustness argument for the primitive.

VII. Related Work

There is an increasing body of work in achieving security properties such as authentication in resource constrained wireless sensor networks. In contrast to the traditional use of asymmetric cryptography [4], [7], [12], [17], Perrig et al present TESLA [9] and its sensor network variant μTESLA [11] that achieves broadcast authentication via key chain hashes. μTESLA achieves asymmetry by delaying the disclosure of symmetric keys. It relies on loose time synchronization and requires dynamic server state to be maintained in the form of precomputed key chains. A sender broadcasts a message with a Message Authentication Code (MAC) generated with a secret key K, which is revealed a short time later. Dependence on time synchronization is necessary because a message is authenticated only when it is received before the key is disclosed. To continuously authenticate broadcast messages, μTESLA divides time period into multiple intervals with a key chain. A message sent at one interval is authenticated when the key is revealed later. Other variants includes multi-level μTESLA [3] that is proposed to efficiently distribute the key chain commitments.

Perrig and Tygar [10] review several other schemes for stream data authentication. For example, Efficient Multicast Stream Signature (EMSS) uses constant distances for the hash links, and MESS uses a randomized method to pick hash links. EMSS/MESS rely on a traditional signature scheme such as RSA [12]. Hash Tree Stream Signature (HTSS) works under an assumption that the entire stream context is known in advance. A Merkle

CC1000 motes	False Positive	False Negative
PL=1	0.94%	0%
PL=9	0.33%	0%

TABLE IX

Testing results for the*CC1000* motes when authentication is done at the packet level. PL=Power Level.

CC2420 motes	False Positive	False Negative
PL=2	0.053%	0%
PL=3	1.05%	0%

TABLE X

Testing results for the *CC2420* motes when authentication is done at the packet level. PL=Power Level.

hash tree is built over all the messages which are authenticated by information associated in the hash tree.

Some other approaches have been proposed especially for secure network programming recently [6], [8]. Both Sluice [8] and Secure Deluge [6] use digital signatures for authentication. The key difference between these two protocols is the granularity of the authentication. Sluice verifies hashes at the page-level while Secure Deluge checks at the packet-level.

Researchers have attempted to use RSSI for localization. It is widely believed in the sensor network community that RSSI is not a good indicator for nodes' location. In this work, we never try to localize a node. By the same token, our work is different from the secure verification of location problem [**?**]. Instead, we use RSSI/LQI to identify a node. Our spatial signature, induced by the radio communications of a node in its neighborhood, is not only related to a node's location but also its inherent physical properties, as we see in the experimental study above. Our work is different from [**?**] by establishing the concept of spatial signature and its realization using statistical modeling. Crypto-free wireless secure communication has received hardware-level attention [**?**], where the fingerprint is induced by hardware variance. Our spatial signature is much simpler and robust since (1) the samples (RSSI/LQI) come for free during the wireless communications (2) it is much harder to fool a neighborhood than a single node.

VIII. Discussion and Conclusion

We have investigated the concept of spatial signature for crypto-free authenticated communication, and proposed a lightweight primitive that realizes the concept in wireless sensor networks. The proposed primitive enables a relatively simple cooperative defense protocol to authenticate the sender of a message, and this protocol in turn enables simple protocols for broadcast authentication, unicast authentication, integrity and non-repudiation of communications. It is lightweight in that it involves efficient local computation, limited communication overhead and relatively small memory utilization. It is scalable in that the overhead does not grow as the density of or the number of motes grows. It is compatible with commonly available network stacks, in that its incorporation assumes only the existence of a neighborhood service. It is also readily combined with other security services to enhance the security level of a network. It is resilient to mote compromise and to mote collusion, and even in the worst case the impact of compromised motes in a region is contained to only their interference area. It provides availability, by not being vulnerable to denial of service attacks.

We have evaluated our primitive on both the *CC1000* motes and the *CC2420* motes. The results are promising, especially on the *CC2420* motes. Typically, we find that on a plane only two neighbors (as opposed to the three suggested by Proposition 3.1) suffice. The stability of RSSI renders it a more attractive basis for realizing our primitive, and will likely only grow with improvements in radio hardware, with consequent improvement in the accuracy of our primitive.

We have not carried out experiments with traffic patterns such as convergecast traffics. One reason is that the accuracy of the primitive is mainly due to the fidelity of the features rather than the pattern of the traffic. We intend to, however, consider the impact of other traffic models in future work.

Also, in this particular evaluation, we have not realized the model using mixtures of gaussian functions, but have used only a one-degree normal function for simplicity. The use of mixtures to simulate more complicated pattern of features is worthy of careful consideration.

Note that conventionally motes allow only RSSI/LQI to be acquired once per message. As

mentioned above, we have opted to obtain multiple samples per packet to enable robust packet level validation and have modified the radio component on the *CC1000* motes accordingly. For the $CC2420$, we note that during a packet reception, an RSSI value is generated every few bytes. This is the basis for our current effort in modifying the *CC2420* motes.

In the task of batch level testing, one possible problem is that an adversary may launch preplay messages to disrupt the communication. For example, suppose the batch size is 10, and an attack may inject a few messages within the batch. To ensure the data integrity, one could apply a hash function to append an integrity check value (ICV) to the end of a batch.

So far, we have not considered directional communication in the protocol. If an adversary can use directional communications at any desired power level, it is possible that no mote other than the receiver sees the message. In this case, some consensus on the receipt time at the neighboring motes appears to be needed. One way of making this efficient could be to send the receiver's receipt time in an acknowledgement message. If any of the other neighbors notice their time is different, they can report the anomaly. We consider the problem of directional communication as our future work.

Another attack to consider is where the adversary starts jamming the protocol immediately after it impersonates a sender. Adding an acknowledgement from the receiver to our cooperative defense protocol can deal with such an attack. Other future work includes consideration of crypto-free alternatives for other security services, including privacy.

REFERENCES

[1] http://www.tinyos.net.

[2] J. A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *Technical report, U. C. Berkeley*, April 1998.

[3] D. Liu and P. Ning. Multi-level μTESLA: Broadcast authentication for distributed sensor networks. *ACM Transactions in Embedded Computing Systems (TECS)*, vol.3(no.4), 2004.

[4] D.Malan, M. Welsh, and M. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. *In First IEEE International Conference on Sensor and Ad hoc Communications and Networks, Santa Clara, CA, USA*, Oct 2004.

[5] J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. *In ACM International Conference on Embedded Networked Sensor Systems*, pages 81–94, November 2004.

[6] P. K. Dutta, J. W. Hui, D. C. Chu, and D. E.Culler. Securing the deluge network programming system. *IPSN'06*, April 2006.

[7] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shants. Comparing elliptic curve cryptography and rsa on 8-bit cpus. *In Workshop on Cryptographic Hardware and Embedded Systems*, 2004.

[8] P. E. Lanigan, R. Gandhi, and P. Narasimhan. Sluice: Secure dissemination of code updates in sensor networks. *In the 26th International Conference on Distributed Computing Systems (ICDCS'06)*, July 2006.

[9] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. The TESLA broadcast authentication protocol. *RSA CryptoBytes*, 5(Summer), 2002.

[10] A. Perrig and J. D. Tygar. Secure broadcast communication in wired and wireless networks. *Kluwer Academic Publishers*, 2003.

[11] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar. SPINS: Security protocols for sensor networks. *in Proceedings of Seventh Annual International Conference on Mobile computing and Networks*, July 2001.

[12] R. Rivest, A. Shamir, and L. Adelman. A method for obtainning digital signatures and public key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[13] C. D. Sheet. http://www.chipcon.com/files/cc2420_data_sheet_1_3.pdf.

[14] K. Srinivasan and P. Levis. RSSI is under appreciated. *Proc. of the Third Workshop on Embedded Networked Sensors, EmNets 2006, Boston, MA*, May 2006.

[15] S. S. Kulkarni and L. Wang. Mnp: Multihop network programming for sensor networks. *In International Conference on Distributed Computing Systems*, pages 7–16, June 2005.

[16] T. Stathopoulos, J. Heidemann, and D. Estrin. A remote code update mechanism for wireless sensor networks. *Technical Report CENS-TR-30, UCLA*, November 2003.

[17] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):74–84, 1976.

[18] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. *ACM SenSys*, 2003.

[19] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. *ACM SenSys*, pages 1–13, 2003.