9

Opening Up the Last Frontiers for Securing the Future Wireless Internet

Wade Trappe, Arati Baliga, and Radha Poovendran

Abstract

Due to the low cost and ease of deployment associated with wireless devices, wireless networks will continue to be the dominant choice for connecting to the future Internet. Beyond serving as an edge-connecting medium, the rapid improvement in communication rates for emerging wireless technologies suggests that wireless networks will also play an increasingly important role in building the backbone of the future Internet. As wireless components become integrated into the design of future network architectures, one significant concern that will arise is whether their pervasiveness, affordability, and ease of programmability might also serve as a means to undermine the benefits they might bring to the future Internet.

Just as the future Internet initiative has brought new perspectives on how protocols should be designed to take advantage of improvements in technology, the future Internet initiative also allows us to reexamine how we approach securing our network infrastructures. Traditional approaches to building and securing networks are tied tightly to the concept of protocol layer separation. For network protocol design, routing functions are typically considered separately from link layer functions, which are considered independently of transport layer phenomena or even the very applications that utilize such functions. Similarly, in the security arena, MAC-layer security solutions (e.g., WPA2 for 802.11 devices) are typically considered as point-solutions to address threats facing the link layer, while routing and transport layer security issues are dealt with in distinct, nonintegrated protocols like IPSEC, TLS, or even in the abundance of recent secure routing protocols.

Although traditional security solutions, that is, cryptographic protocols that work in isolation, are an essential step to understanding how to secure networks, they do not represent a holistic approach. Just as there are significant

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

performance gains to be achieved when combining information from multiple layers to build improved MAC and routing functions,[1] so too is there the potential to significantly improve the security of the future Internet by considering cross-layer approaches to security. The network modality that promises the most opportunities for cross-layer design is wireless. Physical properties outside the normal purview of the network, such as the device itself, or the location of communicating entities, or even the physical properties of the signals being transmitted, can serve as cross-layer information for enhanced security. In this chapter, we will examine the use of cross-layer mechanisms that pull information from the device, from location, and from the physical layer itself to open up the *last frontier* of security design.

9.1 Security Challenges Facing the Future Wireless Internet

Before commencing with this discussion, we briefly describe the potential threats and security opportunities that we envision are possible in the future wireless Internet. As a starting point, we must recognize that wireless devices are inherently commodity items – they are generally low-cost, highly portable, very heterogeneous in their forms, and are becoming increasingly more programmable. One of the great success stories behind wireless networking is that wireless technologies have made networking and communication connectivity available to the broader society. Even the most technically unsavvy person can purchase a wireless router from their local department store for a very accessible price, and deploy their own network, while it requires a far more technically astute person (or team) to deploy and administer a wired router. Not only does this imply that wireless devices are readily available for legitimate purposes, but it also implies that wireless devices could become an ideal platform for illegitimate purposes. This fact, when combined with the fact that wireless devices are small, often hand-held, and allow their users to connect to the broader network anywhere at anytime, means that wireless will be an ideal modality to launch a variety of threats against the broader network and its users. As if this were not a harsh enough scenario, there is a movement to make wireless devices increasingly programmable. Already there are a handful of programmable smart phones, such as the Google Android[2] and Apple iPhone[3] and supporting SDKs[4,5] that promise to make it easier to develop new software for good and bad purposes. At the same time, new radio platforms, like software-defined radios and cognitive radio (CR) platforms, are being developed to open up the lower layers of the protocol stack for general development. Consequently, many threats that might have been prevented are now easily possible because firmware restrictions are no longer in place.

We may decompose the threats facing the future wireless Internet in terms of a classical CIA (confidentiality, integrity, and availability) framework. In

the discussion that follows, we list several CIA threats made possible by the commodity nature of wireless technologies.

- Confidentiality: Wireless communications between entities are especially susceptible to confidentiality threats from attackers interested in snooping over the message contents. As messages are broadcast over the air, malicious adversaries can easily intercept and interject packets. For unencrypted communication, an adversary can easily decipher packet contents by listening to broadcast packets, consequently violating confidentiality. Alternatively, man-in-the-middle attacks are possible by injecting false packets, thereby allowing an adversary to decipher traffic crossing the network.
- Integrity: Due to the commercial nature of wireless devices, integrity needs to be established at various levels. This involves integrity of the wireless device itself, integrity of the software running on the device, and integrity of message communication between the sender and the receiver. Attacks can be carried out at different levels. For example, an attacker can manipulate the device hardware to alter its behavior. He can alter the device software to install malicious versions of system and application software. In case of cognitive radios, a malicious attacker may tamper with the installed policies while maintaining unaltered version of the software. Finally, in systems that do not verify message integrity, malicious adversaries can inject false messages or carry out man-in-the-middle attacks.
- Availability: More malicious attacks could involve a user programming a CR device to give him/her advantage relative to neighboring devices. For example, a greedy user might seek to decrease the back-off window size in an 802.11 implementation, and as a result obtain a larger fraction of the channel utilization. Generally, such greedy attacks can take a variety of forms, ranging from bypassing agreed-on MAC-layer behavior to ignoring implementations of fairness in spectrum-etiquette policies. A deleterious adversary might seek to turn the CR platform into a jamming platform by listening to channel utilization and emitting short blocker packets to prevent the reception of packets.

9.2 The Final Frontier: Introducing the Physical into Security

The traditional approach to security involves layer-specific protocols that are unaware of the platform or the physical medium on which their associated messages rely. Throughout this chapter, we take the view that the *physical* world represents an important aspect of communication that must be addressed in order to properly have a holistic approach to securing devices on the future Internet. By physical world, we mean the physical platform associated with the

communication, the physical medium over which communications are carried, and also the physical context of the communication in terms of the locations of the communicating entities.

We now highlight several types of new security services that may be built by using the physical aspects of communication into the security framework.

- Physical Device Integrity Services: The programmability allowed in cognitive radios necessitates an architecture that does not allow the devices to violate high-level spectrum etiquettes. The physical devices integrity services should be able to verify the integrity of the physical device and the policy enforcement code that runs on top of it.
- Authentication/Identification Services: The uniqueness of the channel between two locations provides a means for uniquely identifying wireless entities. Devices may authenticate themselves based on their ability to produce an appropriate received signal at the recipient.
- Confidentiality Services: The fact that pairwise radio propagation laws between two entities are unique and decorrelate quickly with distance can serve as the basis for establishing shared secrets. These shared secrets may be used as encryption keys for higher-layer applications or wireless system services that need confidentiality.
- Availability Services: RF-specific denial-of-service attacks targeting the ability of radio devices to transmit or receive messages may be launched against wireless networks. Detecting RF interference, or jamming, attacks must be performed at lower layers. Spectral evasion strategies may be integrated into the devices so as to assure the availability of the wireless network in the presence of interference attacks.
- Verifiable Location Services: Radio communications do not exhibit brick-wall propagation, and consequently wireless networks may be accessed from locations other than their intended coverage region. This phenomena can facilitate threats to the security of both wireless networks and the broader Internet. However, we may also use lower-layer functionalities, such as power control, to provide mechanisms to verify the location of mobile entities.
- Non-repudiation Services: RF energy naturally radiates, and wireless entities within the radio coverage pattern may serve as witnesses for the actions of the transmitter. This makes it harder for radio entities to deny receiving a message or having performed an action. We may introduce communication auditors into the wireless infrastructure to assist in quantifying the trust of wireless entities.
- Forensic Services: The wireless medium is perturbed by the introduction of new entities, whether physical objects or other radio transmitters. Lower-layer

information can serve as forensic evidence for detecting an environmental change and possibly even identifying the cause of such a change. Wireless forensic services identify unauthorized intrusions in the radio environment and serve to actuate responses, such as adjusting system level security policies.

In the rest of this chapter, we shall examine potential physical security mechanisms that operate at the device level, take advantage of location contexts, and use information from the physical layer.

9.3 Platform and Device-Level Assurance

A starting point for protecting against attacks is to realize that if all wireless devices were following their proper hardware and software instructions, then no attacks would be present. Consequently, all attacks originating from wireless (or other) devices originate from devices that are executing their supposed functions improperly. For example, the software associated with networking functions may have been altered, and such malicious code may then be used to subvert the network. Malicious code may corrupt routing updates, selectively drop messages, mount slander attacks, and allow nodes to collude to hurt other nodes.

As a first line of defense, it is natural to attempt to verify that the code running on a network node is approved. Checking the integrity of hardware and software thus can act as a first filter in preventing attacks. Typically, the research in this arena has pursued two different directions: hardware- and software-based attestation techniques. On the hardware front, trusted platform modules (TPMs) have been used to establish a dynamic root of trust, and hardware protection can be used to prevent unauthorized access to the secure loader block, identify whether code execution occurs after a reboot, and allow for code to be executed in an isolated environment.

Recent research[6] has shown that hardware-based mechanisms can provide a powerful abstraction to implement dramatically improved secure network protocols. The basic premise is that if one can trust the code that has generated an output, and further that this code includes input verification, then the output can be trusted. This new approach for designing secure networking protocols promises to greatly enhance the security and efficiency of distributed systems, and will be an important component to securing the future wireless Internet. Unfortunately, advancements in hardware-based attestation cannot address security threats being conducted from devices that do not employ TPMs. In such a case, software-based code attestation can be used. Recently, the SWATT and Pioneer systems have shown that it is possible to provide software attestation on legacy platforms.[7,8]

Building on such work, we now explore how trusted platform technology may be used to ensure that future programmable wireless platforms, like the cognitive radio, exhibit trustworthy behavior.

9.3.1 Security and Cognitive Radios

There has been considerable effort directed at developing "cognitive radio" (CR) platforms, which will expose the lower layers of the protocol stack to researchers and developers.[9] This initiative is supported by two separate technical efforts: First is a wealth of research devoted to uncovering the gains that are possible by letting the lower protocol layers become programmable and adaptable; second are the recent advances in programmable integrated circuits that have significantly increased the amount of computation that can be done without requiring specialized hardware/firmware components. By being able to scan the available spectrum, select from a wide range of operating frequencies, adjust modulation waveforms, and perform adaptive resource allocation – all of these in real time – these new "cognitive" radios will be able to adapt to a wide variety of operational settings, supporting a true "anywhere-anytime" vision of the future Internet.

Although there is great potential for such a radio platform, some caution regarding their ubiquitous use in wireless systems is warranted, because their deployment will not be limited to the laboratory. Already, the GnuRadio platform[10] is available for general use, and supporting this platform is an open-source software effort to develop GnuRadio "blocks"[11] – software modules capable of conducting a broad range of functions associated with the reception/transmission of radio signals. Other CR platforms, such as the Xilinx-based Rice WARP cognitive radio platform[12] and the WINLAB WiNC2R platform,[13] will also reach a large consumer base with similar open-source efforts supporting lower-layer protocols.

The open-source nature of cognitive radio software is empowering but also dangerous. It is easily conceivable that inexpensive and widely available cognitive radios could become an ideal platform for abuse since the lowest layers of the wireless protocol stack are accessible to programmers. Thus, the gains promised by adaptive resource-allocation schemes and good spectrum-etiquette policies can be negated if cognitive radio devices can be reprogrammed to violate or bypass locally fair-spectrum policies either maliciously or inadvertently. If fail-safe mechanisms are not employed, individual devices could use the wireless medium to their advantage at the expense of the greater good.

To regulate this future radio platform, we present a framework, known as TRIESTE (Trusted Radio Infrastructures for Enforcing SpecTrum Etiquettes), which can guarantee that a cognitive radio behaves according to acceptable communal policies.[14]

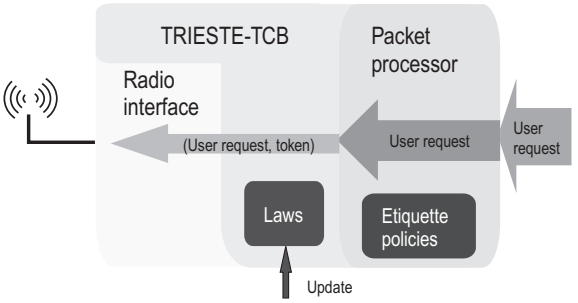

Figure 9.1. The architecture of the Cognitive Radio with on-board TRIESTE-TCB.

We begin by assuming the presence of a third party, known as the Spectrum Law Makers, which give general guidelines on how the cognitive radios should operate. For example, the Federal Communications Commission (FCC) might be such an entity, and would provide rules describing how spectrum should be accessed, which bands are not allowed to be transmitted on, and requirements on interference between cognitive radios and other *primary* wireless modalities. To enforce these rules, we believe it is necessary to have an on-board Trusted Computing base/module (TRIESTE-TCB) in each cognitive radio that enforces the spectrum laws and etiquettes.

The TRIESTE-TCB, as depicted in Figure 9.1, includes all the hardware and software in the cognitive radio that enforces universal laws and etiquette policies passed down by the Spectrum Law Makers. The TRIESTE-TCB can be thought of as a control gate that user processes have to go through to access the radio. In TRIESTE, typically, before the user can transmit information over a certain radio spectrum band, the user/process would send a spectrum access request, which includes information about the target radio frequency band, the spectrum etiquette the user will follow, the transmission power, transmission duration, and so forth, to the packet processor. Here, we note that we shall abuse terminology and, for simplicity, collectively refer to the packet processor as an entity consisting of multiple processors handling packets, such as the Network Processor, the CR Policy Processor, and so on. The packet processor shapes the user's radio access request according to the spectrum-etiquette policies programmed by the user or spectrum owner, then passes the modified user request to TRIESTE-TCB. The TRIESTE-TCB in turn will validate the request against the laws available to it and will allow the request to go through only if it does not violate any of those laws.

The TRIESTE-TCB would evaluate the access request along with the user's credentials and checks it against the spectrum laws. If the request and credential combination is valid in the context of spectrum laws, then the TCB would issue a privilege token for that request. The privilege token is a tuple consisting of the ⟨spectrum-access-details, timestamp and a signed hash of

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

[spectrum-access-details‖timestamp]⟩. The spectrum-access-details might specify, for example, the radio frequency, duration, and spectrum access limitation granted. If the user's credentials do not permit the privilege level of the request or if the combination somehow violates some spectrum law, then the TRIESTE-TCB could either try to find a permissible modification of the request that is in compliance with the spectrum laws or reject the request if such a modification is not feasible. We note that the user's credentials may change over time, and each request would be evaluated in the context of the credentials presented with it. For example, a user with emergency-responder credentials would have higher privilege spectrum access during an emergency situation as opposed to during a nonemergency one.

TRIESTE-TCB would compare the access request with the spectrum laws, and only if the request does not violate any spectrum laws would the request be validated and update-privilege tokens issued to the user, otherwise the request will be rejected or modified. An access token, which specifies the radio frequency, duration, and spectrum access limitation granted, together with the user request, will be passed to the radio interface processor via a tamper-proof path. Typically, to further prevent a user from bypassing the TRIESTE-TCB and forging the token itself, authentication mechanisms are necessary to assure the token is granted by the TRIESTE-TCB.

Inside TRIESTE-TCB would be a monitoring component, known as the monitor verifier, which will monitor the on-board radio activity and observe the radio environment, and check any potential violation by comparing "spectrum laws." If the user does not follow the rules it claims to obey, the TRIESTE-TCB will stop the radio operation and revoke the user's token/privilege.

We now discuss a few challenging issues on which the TRIESTE-TCB could depend. First of all, the law should be stored in a secure storage to ensure protection against tampering. Additionally, after the token has been issued to the user, the association relationship of user request and token should not be altered as the "(user request, token)" pair passes between cognitive radio components. To achieve integrity, encryptions can be used, though at the cost of additional computational overhead. Alternatively, we can design the cognitive radio in such a way that after the creation of "(user request, token)," the pair travels among components via trusted paths. Thus, the data pair cannot be intercepted on the way, nor can the content of the user request be changed. As the spectrum laws will evolve over time, it is thus desirable to make the law extendable.

Since the cognitive radio is a programmable wireless platform that will support a wide range of radio network scenarios, from autonomous agile radios to those that use higher-layer protocols to share spectrum, it is wise to consider a generic high-level architecture, such as shown in Figure 9.2. Here, a CR consists of Flexible RF units, a baseband processor, a network processor, and a cognitive radio policy processor (which also functions as the host). Besides those components, we have added a logical component, the TRIESTE-TCB, to enforce

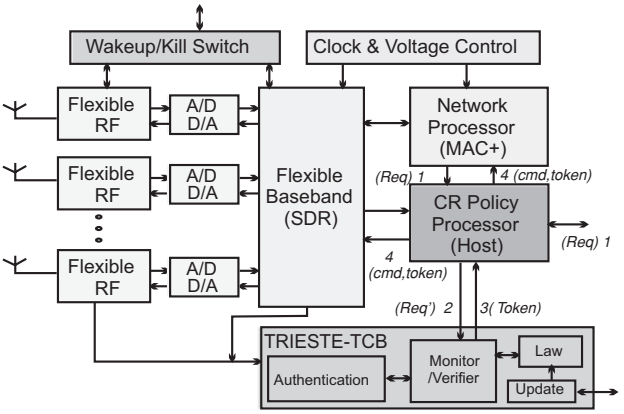

Figure 9.2. A generic SDR/CR platform involving RF processors, baseband processor, network processor, and the cognitive radio processor. Note that TRIESTE regulates via a TCB component and an externally accessible authenticated kill-switch.

the spectrum laws. Here we want to point out that the law/policy enforcement activities are likely to be performed at several functional places within the CR, because law/policy enforcement is potentially related to every network protocol that will access the spectrum. Although we show the TRIESTE-TCB in one monolithic block, in implementation, the functions of the TRIESTE-TCB will be located in firmware in different processors.

As noted earlier, the TRIESTE-TCB can be thought of as the controlled gate that users have to go through to access radio. The basic structure of TRIESTE-TCB consists of a generic *Controller* that can interpret and enforce any well-formed *Law*. As we pointed out earlier, the TRIESTE-TCB is a virtual block, and the real functions of the TRIESTE-TCB will be located in hardware or software on different components of the CR. In particular, many of the proposed functionalities of the TRIESTE-TCB might require a secure, tamper-proof chip on board the CR platform that is dedicated to providing a hardware-based root of trust. Recent efforts by the Trusted Computing Group have mapped out specifications for the Trusted Platform Module, to enable trusted computing functions such as platform attestation/integrity, hardware-based cryptographic functionality, and secure storage.[15] Manufacturers, such as Atmel, have already produced TPM chips that find use in digital rights management services, and such technologies warrant application to securing CRs.

In the TRIESTE framework, cognitive radios must adhere to the Spectrum Laws published by agencies, such as the FCC. Future cognitive radios should be able to adapt to new laws/policies dynamically, as laws/policies tend to change over time. A starting point for defining such laws would be to use XGPL (XG Policy Language)[16] to express spectrum policies formally. XGPL is part of the XG (neXt Generation Communications) research program that aims to let radios

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

utilize available spectrum intelligently and dynamically based on the knowledge of actual conditions and spectrum policies. In particular, the XG project chose OWL (Web Ontology Language) as its XG Policy Language for several reasons. First of all, OWL provides the structure and richness needed to express policies. Secondly, general theorem proving/reasoning engines for deductive inference are already available. Finally, OWL is an efficient language for describing data and passing it around different systems.

OWL is originally designed for processing information on the Web and is designed to be interpreted by computers. It is written in XML (Extensible Markup Language). We note that OWL is not another programming language, but is a structured way to build representations for information and policies for machine understanding. For example, the OWL expression of magnitude 10 is as follows:

```
<xgparam:magnitude>
    <xsd:integer rdf:value="10" />
</xgparam:magnitude>
```

The paragraph above defines a property "magnitude" in the name space "xgparam." The value of the property magnitude is 10, the type of the value is integer, which is defined in namespace rdf. More detailed and precise exposition on OWL can be found in OWL Web Ontology Language Guide.[17]

For the remainder of this chapter, we use the shorthand notion described in XG Working Group Document.[16] The shorthand notation yields representations equivalent to OWL representations. For example, we describe the previous "magnitude is 10" in the following way:

```
(magnitude 10)
```

Detailed mapping from OWL to shorthand notion can be found in XG Working Group Document.[16]

A spectrum policy rule is composed of three facts: a selector description, an opportunity description, and a usage constraint description, as shown below:

```
(PolicyRule (id Policy_name)
    (SelDesc S)
    (OppDesc SomeOpp)
    (UseDesc SomeUseDesc)
```

The first part in a spectrum policy rule is a selector description, which is used to filter policy rules to the subset of rules that may apply to a given situation. The selector description contains one or more facts that describe the frequency, time, and region the policy covers, the authority that defines the policy, and the radio device to which the policy rule applies. For example, a selector description may include filters such as "applies to operation in U.S.A" or "applies to operations in the 3.6 GHz to 3.7 GHz bands."

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

The second part in a policy rule is an opportunity description, which is used to evaluate whether the transmission request is valid or not based on whether or not a given environment and device state match the opportunity description in the filtered subset rules. For example, the opportunity description can be "if a beacon is heard at 823 *MHz*," or "peak received power is less than −80 *dBm*."

A valid opportunity indicates transmission that conforms to the usage constraint description is permitted. Usage constraint description constrains the radio behavior, such as "transmit with a maximum power of −10 *dBm*" or "maximum continuous on-time must be 1 second and the minimum off-time must be 100 *msec*."

We envision that usually, a spectrum policy rule is first defined in XGPL, then each element (a selector description, an opportunity description, and a usage constraint description) is defined in a format similar to the format used to specify policy rules.

The spectrum law includes both spectrum access laws and punishment laws. We have discussed how to express spectrum access rules using XGPL. In the original XG project, XGPL is designed to describe spectrum access control. XGPL was not used to specify any form of punishment for spectrum abuse. In particular, the underlying idea of the XG project is that the regulatory policy does not tell the radio what to do; it only defines what constitutes authorized use of the spectrum. Punishment, however, tells the radio what should be done once violation has occured.

We believe that it is necessary to define punishment rules as part of the spectrum laws, because punishment can serve to prevent potential spectrum violation as well. Although it might be a challenge, XGPL can be extended to define punishment rules. One way to define punishment is to add one more description, the punishment description, into the policy rules as shown:

```
(PolicyRule (id Policy_name)
    (SelDesc S)
    (OppDesc SomeOpp)
    (UseDesc SomeUseDesc)
    (PunDesc SomeAction)
```

One possible way to perform the punishment is as follows. If the punishment rule is selected and activated, then new punishing rules with certain expiration period will be generated based on the level and type of punishment, and inserted into the existing spectrum polices for specified amount of time. For example, the newly generated punishing spectrum access rules could be that the radio device cannot access to band 3.6−3.7 *GHz* for two hours. Of course, precedence mechanisms are needed to resolve conflict. Detailed techniques for defining punishment and precedence require further investigation.

One concern regarding a TPM-based approach to building the TRIESTE-TCB is that TPMs are generally focused on software rather than hardware attacks, and simple hardware-based man-in-the-middle attacks can compromise the boot sequence. Because this attack does not use TPM die probes, the vulnerability is not overcome with stronger chip-level tamper resistance. As a consequence, hardware-related security challenges for CR include: (1) deductions made in the software layer may no longer hold when the hardware layer is accessible, and (2) hardware-protected information may not necessarily be localized to a single TPM chip. Absolute physical protection of integrated circuits is difficult because testing is required after packaging. The state-of-the-art in tamper and probing resistance involves proprietary commercial techniques. Regardless of the physical protection methods employed, combining as many functions as possible on one chip is desirable because it increases the cost of a physical attack (since external pin probes may be insufficient) and decreases the cost of protection (since fewer chips need to be tamper-resistant). Current trends in CR design suggest that the most suitable platform design involves FPGAs, which have the needed adaptability and logic capacity for CR functions. Security aspects of the interfaces and functionality assigned to various CR components, and the FPGA in particular, are new system-level partitioning constraints that need to be developed. In the hardware domain, the design principles used to improve performance, namely nonsharing of computation, communication, and memory resources, also promote system security by restricting access to private information. Hardware-based access restrictions are generally simpler to assure than software or software-managed hardware (such as memory management units). These guarantees are diminished when the hardware is shared between different processes, because cached private information often exists prior to a context switch. A further area of investigation is the enforcement of basic operational policies using hardware-layer "interlocks" that cannot be overridden by software layers. This would require analyzing the interfaces and dependencies between hardware and software layers, selecting the policies to be enforced with hardware, formal state analysis of the hardware blocks responsible for policy enforcement, and a mechanism for securely updating policy enforcement circuits.

9.4 Location as an Enabler for Security Services

Radio signals in wireless networks may be accessed from locations other than their intended coverage region. This fact poses several security threats to the deployed wireless networks because it can be accessed by malicious users from outside the perimeters. Therefore, location information and position verification methods are crucial to the deployment of a security framework, which can provide different types of access control policies depending on the physical

location of the device. However, we may also use lower-layer functionalities, such as power control, to provide different kinds of security services in wireless networks. In this section, we focus on two different ways in which location can be used to enhance security: First, we examine the use of location as a means to detect the spoofing of a wireless entity; and, second, we examine a key management scheme that uses location information and power control to allow for secure multicast in wireless ad hoc networks.

9.4.1 Location-Based Recognition of Spoofing Attacks

Spoofing attacks are serious threats because they can facilitate a variety of traffic injection attacks against networks. These attacks are particularly easy to conduct at the edge of the Internet, where wireless devices, such as sensor nodes and wireless LANs, cannot employ appropriate authentication mechanisms to detect the injection of false messages. It is thus desirable to detect the presence of spoofing and eliminate them from the network. The traditional approach to address spoofing attacks is to apply cryptographic authentication. However, authentication requires additional infrastructural overhead and computational power associated with distributing and maintaining cryptographic keys. Due to the limited power and resources available to wireless devices on the edge, it is not always possible to deploy authentication. In addition, key management often incurs significant human management costs on the network.

We will now examine how the physical properties associated with where wireless transmissions are being sent from can be used to detect spoofing. Specifically, we present a scheme for both detecting spoofing attacks and localizing the positions of the adversaries performing the attacks. The approach that we summarize utilizes the Received Signal Strength (RSS) measured across a set of monitoring nodes (e.g., access points) to perform spoofing detection and localization.

9.4.1.1 Formulation of Spoofing Attack Detection

In a spoofing attack on a wireless (edge) network, an adversarial node will claim the identity of another, legitimate node (e.g., by altering its MAC address).[18] Unless the adversary is located at precisely the same location as the legitimate node, it should be possible to distinguish between the two communication streams by localizing each transmission and noticing that packets coming from the claimed address appear to come from multiple, simultaneous locations.* There can be multiple nodes spoofing the same MAC address.

* We note that the methods that we described are most suited for scenarios where the legitimate entity is present at the same time as the adversarial entity.

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

RSS is a physical parametes, widely available in deployed wireless communication networks, and is intimately tied to the location of a device in physical space. For this reason, RSS is a common physical property used in localization algorithms,[19–21] and can be used to detect communication spoofing.

Spoofing attack detection can be formulated as a statistical significance test where the null hypothesis is:

$$\mathcal{H}_0 : \text{normal (no attack).}$$

In significance testing, a test statistic $\mathbf{T}$ is used to evaluate whether observed data belongs to the null hypothesis or not. If the observed test statistic $\mathbf{T^{obs}}$ differs significantly from the hypothesized values, the null hypothesis is rejected and we claim the presence of a spoofing attack.

Although affected by random noise, environmental bias, and multipath effects, the RSS value vector, $\mathbf{s} = \{s_1, s_2, \ldots s_n\}$ (n is the number of landmarks/access points [APs]), is closely related to the transmitter's physical location and is determined by the distance to the landmarks.[21] We will describe the collection of vectors $\mathbf{s}$ as constituting a signal space. When there is no spoofing, for each MAC address, the sequence of RSS sample vectors will be close to each other and will fluctuate around a mean vector. However, under a spoofing attack, there is more than one node at different physical locations claiming the same MAC address. As a result, the RSS sample readings from the attacked MAC address will be mixed with RSS readings from at least one different location. Based on the properties of the signal strength, the RSS readings from the same physical location will belong to the same cluster points in the n-dimensional signal space, whereas the RSS readings from different locations in the physical space should form different clusters in signal space.

This observation suggests that we may conduct cluster analysis on the RSS readings from each MAC address to detect spoofing. For example, the K-means algorithm is an easy-to-use and efficient candidate algorithm for clustering. If there are M RSS sample readings for a MAC address, the K-means clustering algorithm partitions M sample points into K disjoint subsets S_j containing M_j sample points so as to minimize the sum-of-squares criterion:

$$J_{min} = \sum_{j=1}^{K} \sum_{\mathbf{s_m} \in S_j} \|\mathbf{s_m} - \mu_j\|^2 \tag{9.1}$$

where $\mathbf{s_m}$ is a RSS vector representing the mth sample point and μ_j is the geometric centroid of the sample points for S_j in signal space. Under normal conditions, the distance between the centroids should be close to each other because there is basically only one cluster. Under a spoofing attack, however, the distance between the centroids is larger because the centroids are derived

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

from the different RSS clusters associated with different locations in physical space. We thus choose the distance between two centroids as the test statistic **T** for spoofing detection,

$$D_c = ||\mu_i - \mu_j|| \tag{9.2}$$

with $i, j \in \{1, 2..K\}$.

The thresholds used in defining the critical regions of the significance test can either be set empirically or via an analytical model. To illustrate, we use the following definitions: *an original node* P_{org} is referred to as the wireless device with the legitimate MAC address, while *a spoofing node* P_{spoof} is referred to as the wireless device that is forging its identity and masquerading as another device. We now present results from an experimental validation that shows that position information can be a valuable tool for detecting spoofing.

We will evaluate the performance of a cluster-based spoofing detector by analyzing the resulting detection rate and false-positive rate. The detection rate is defined as the percentage of actual spoofing attack attempts that are correctly classified as being an attack. Note that when the spoofing attack is present, the detection rate corresponds to the probability of detection P_d. Under normal (non-attack) conditions, a detection corresponds to a false alarm, and hence we are also interested in the false-positive P_{fa} rate.

Table 9.1 presents the detection rate and false-positive rate for an 802.11 network and a 802.15.4 network under different threshold settings (details describing the experimental set up can be found in Yang et al. 2009).[22] The results show that for false-positive rates less than 10 percent, the detection rates are above 95 percent. Even when the false positive rate goes to zero, the detection rate is still more than 95 percent for both 802.11 and 802.15.4 networks.

Table 9.1. *Detection Rate and False-Positive Rate of the Spoofing Attack Detector. Two Different Types of Wireless Networks Were Used (802.11 and 802.15.4) to Show the Feasibility of Using Location to Detect an Identity Attack Against a Wireless Network, Without Resorting to Cryptographic Mechanisms*

Network, Threshold	Detection Rate	False Positive Rate
802.11, $\tau = 5.5$dB	0.9937	0.0819
802.11, $\tau = 5.7$dB	0.9920	0.0351
802.11, $\tau = 6$dB	0.9884	0
802.15.4, $\tau = 8.2$dB	0.9806	0.0957
802.15.4, $\tau = 10$dB	0.9664	0.0426
802.15.4, $\tau = 11$dB	0.9577	0

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

9.4.2 Location-Oriented Multicast Key Management

When sending an identical message to multiple receivers, adopting the multicast communication model reduces the network traffic and allows the sender to conserve energy consumed for data processing. Several critical network operations such as routing, neighbor discovery, key distribution, and topology control can benefit from multicast by efficiently distributing protocol status updates or any other required data. Furthermore, in a wireless environment, due to the broadcast nature of the wireless medium, multicasting has the potential to not only reduce the network traffic in number of messages, but also reduce the network energy expenditure. A single broadcast transmission will reach any receiver within the communication range* of the source. However, anyone in range can listen to an information broadcast over the wireless medium. Hence, it is important to ensure that only the intended receivers have access to the group communication at any given time.

Encrypting the information transmitted over the open wireless channel is the most common technique for securing the multicast communication.† The use of cryptography requires all the valid receivers to hold the decryption key in order to decrypt a common message. The shared decryption key is called Session Encryption Key (SEK). To preserve the secrecy of the multicast data, the SEK needs to be updated each time a membership change occurs. For updating the SEK, multicast members share additional keys called Key Encryption Keys (KEK) that allow the secure update of the SEK to valid members. The *key management* problem is to ensure that only the legitimate members of the multicast group hold valid keys at any time during the session. In the presence of group members that may join or leave the multicast group, the key management problem is equivalent to the problem of finding efficient mechanisms to generate, assign, and distribute cryptographic keys. Hence, the key management problem can be reduced to the *key distribution* problem, which addresses the secure and efficient distribution of the cryptographic keys to valid members.

We will show that it is possible to provide energy-efficient key distribution scheme for implementing group access control for multicast communication in wireless ad hoc networks. To reduce the energy expenditure (physical layer parameter) of the key distribution (application layer operation), we propose a cross-layer design approach that incorporates (a) the network topology (location of the nodes) and (b) the propagation medium characteristics (physical layer). We note that the use of network topology for efficient multicast key management

* The communication range is defined as the maximum distance from the transmitter to a receiver, so that the signal-to-noise ratio (SNR) is above the required threshold for communication.

† Additionally, cryptography can also support group access control for dynamic multicast groups through secure management of the cryptographic keys.[23]

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

of cellular networks has been examined in Sun et al. (2002) and Sun et al. (2003).[24,25]

9.4.2.1 Network Model Assumptions

Network generation. The network consists of N multicast members plus the Group Controller (GC) randomly distributed in a specific area. The GC is also randomly placed within the network region. We consider a single-sender multiple-receiver communication model. We assume that all users can act as relay nodes and therefore relay information to any user within the communication range. We also assume that the network nodes have the ability to generate and manage cryptographic keys.

Node location acquisition. For our main analysis, once generated, the nodes of the network are assumed to be in a fixed location. We also assume that nodes have a mechanism to acquire their location information. Such information is often obtainable through the Global Positioning System (GPS).[26]

However, in many cases, GPS may not be available due to the expensive hardware required (e.g., sensor networks) or the lack of obstacle-free communication (indoor setting). Several approaches have been proposed for acquiring location information without a GPS receiver.[19,27,28,29,30] After a node correctly acquires its location, it can report it to other nodes through a location service algorithm.[31,32] However, to prevent denial-of-service attacks, the use of a secure location verification algorithm is important.[33]

Network initialization. We assume that the network has been successfully initialized and initial cryptographic quantities (pairwise trust establishment) have been distributed through secure channels.[34,35] We further assume that the underlying routing is optimized in order to minimize the total energy required for broadcast. Although it is known that finding the optimal solution for power-optimal broadcast is NP-complete,[36,37] several heuristics resulting in routing trees with satisfactory performance have been proposed in the recent literature.[38,36,39]

Wireless medium and signal transmission. We consider the cases of a homogeneous and heterogeneous medium separately, because the complexity and inputs of the algorithms that we propose differ depending on the type of the medium. In the case of the homogeneous medium, we assume that the transmission power $P(d_{i,j})$ required for establishing a communication link between nodes i and j is proportional to a constant exponent (attenuation factor γ) of the distance $d_{i,j}$, that is, $P(d_{i,j}) \propto d_{i,j}^{\gamma}$. For simplicity, we set the proportionality constant to be equal to 1. An example of a homogeneous path loss medium is an obstacle-free, open space terrain with line-of-sight (LOS) transmission.

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

For a heterogeneous medium, no single path loss model may characterize the signal transmission in the network deployment region. Even when node locations are relatively static, path loss attenuation can vary significantly when the network is deployed in mountains, dense foliage, urban region, or inside different floors of a building. We consider the following two models of varying path loss for calculating the power attenuation at a distance d from the transmitter:[40] (a) suburban area – a slowly varying environment where the attenuation loss factor changes slowly across space; (b) office building – a highly heterogeneous environment where the attenuation loss factor changes rapidly over space.

Antenna model. We assume that omnidirectional antennas are used for transmission and reception of the signal.[38] The omnidirectionality of the antennas results in a property unique in the wireless environment known as the *broadcast advantage*.[38] When the sender transmits a message to a node, any other nodes that lie within the transmission range can receive the broadcasted message for free. Hence, when an identical message needs to be sent to multiple receivers, the sender can significantly reduce the energy expenditure by directly transmitting the data to the farthest member. However, omnidirectional antennas require more power to transmit a signal at distance d than directional ones.[40] We further assume that signal transmission is the major component of energy expenditure and ignore any energy cost due to computation and information processing.[34,41]

Our aim is to develop energy-efficient key management scheme for wireless ad hoc networks, which accounts for the following factors: (a) A_1 : The network topology, that is, node location and relative position to minimize the energy consumption; (b) A_2 : The characteristics of the medium where the network is deployed (path loss parameter, homogeneous as well as heterogeneous, or power measurements); and (c) A_3 : Scalability of communication overhead in both bandwidth and required key storage space with respect to group size N.

9.4.2.2 A New Evaluation Metric for Measuring the Communication Overhead of Key Management

In order to incorporate the features A_1 to A_3 into the key management scheme, we first define a suitable performance evaluation metric that reflects the energy expenditure associated with the key distribution overhead. We then show that if key graphs are evaluated with the new metric, their performance is dependent not only on key graph structure, but also on node location and medium type (homogeneous or heterogeneous).

In wired networks, the communication overhead associated with the key management is measured as the number of messages sent by the GC to the group members in order to complete a key update. In key trees, a higher-degree tree requires a larger number of messages to be transmitted by the GC in case of a

member leave.[42] For logical key hierarchies as proposed in Wang et al. (2000),[42] the communication cost to update the SEK and compromised KEKs after a member deletion is equal to $(\alpha \log_\alpha N - 1)$, where α is the degree of the tree. In Canetti et al. (1999),[23] the authors propose the use of key trees in conjunction with one-way functions, to reduce the communication cost to $(\alpha - 1)\log_\alpha N$ messages per member deletion. None of the metrics that have been developed for the wired networks, with the exception of time delay, is calculated collectively on the whole network. We propose a metric that calculates the energy expenditure of the whole network, occurring due to member leave/deletion.

In wireless ad hoc networks, each message has an associated energy cost that depends on the location of the receiver relative to the source, the routing path that connects them, and the path-loss model assumed. Since in an ad hoc setup, messages with different recipients have different energy requirements, a small number of transmitted messages by the GC does not necessarily translate to low energy expenditure. Hence, a higher-degree tree may have a lower energy cost compared to a lower-degree tree, even if more messages need to be transmitted in the case of the tree of higher degree to update compromised keys after a member deletion. To capture the energy dimension of the key management, we propose a new performance metric called *Average Update Energy*, as defined below.

Definition: Average update energy: Let $\tilde{E}_{M_i}$ denote the energy expenditure for updating the compromised keys after the deletion of the i^{th} member. Also, let $p(M_i)$ denote the distribution of the member leaves/deletions from the multicast group. Then, we define the average update energy required for key update after a member leave/deletion as:

$$E_{Ave} = \sum_{i=1}^{N} p(M_i)\tilde{E}_{M_i} \tag{9.3}$$

We define the update energy in the average sense, since a member deletion triggers transmissions to different subgroups of the multicast group. Hence, $\tilde{E}_{M_i}$ depends on the member that is being deleted. For example, in Figure 9.3(a), if member M_1 were to be deleted, the messages that need to be transmitted are shown in Figure 9.3(b). To further reinforce the idea, if M_8 were to be deleted, the following message transmissions have to take place:

$$\begin{aligned} GC \rightarrow M_7: &\quad \{K'_{2.4}\}_{K_{3.7}}, \{K'_{1.2}\}_{K_{3.7}} \\ GC \rightarrow \{M_5, M_6\}: &\quad \{K'_{1.2}\}_{K_{2.3}} \\ GC \rightarrow \{M_5 - M_7\}: &\quad \{K'_0\}_{K'_{2.1}} \\ GC \rightarrow \{M_1 - M_4\}: &\quad \{K'_0\}_{K_{1.1}} \end{aligned}$$

Note that the member M_7 will receive keys $K'_{2.4}$, $K'_{1.2}$ both encrypted with key $K_{3.7}$. Although the GC can concatenate both keys into one message and

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

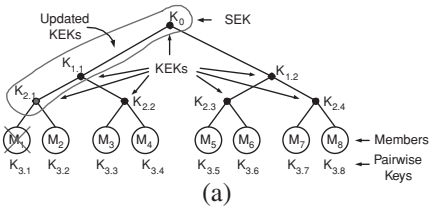

Figure 9.3. (a) A binary logical hierarchical key tree. Members are placed at the leaf nodes. Each members holds the keys traced along the path from the leaf to the root of the tree. If M_1 leaves the multicast group, all keys known to it (keys traced along the path from the leaf $[M_1]$ to the root of the tree) are updated (b) Update messages sent by the GC after M_1 leaves the multicast group.

update both keys to M_7 with one transmission, we show that two messages are transmitted from the GC to M_2. We intentionally note the messages separately for counting purposes. Assuming that all the keys have the same length and the concatenated message is twice as long when compared to a single message, the two representations are equivalent in both bits transmitted and energy consumed by the network. A key concatenation has the advantage of guaranteeing that both keys are received with same delay by M_7. Sending the keys through separate messages is suitable in a highly lossy medium where frequent retransmissions occur.[43] Because our scheme is concerned with the energy consumption, we use a representation that allows us to count the amount of energy spent, using the parameter of one new key encrypted per message.

From these two examples, for an ad hoc network with random node distribution, $\tilde{E}_{M_1} \neq \tilde{E}_{M_8}$, that is, the deletion of M_1 and M_8 result in different energy expenditures. As mentioned earlier, we consider the case of a member leave, because significantly higher communication cost occurs during a member leave than a member join.[44] We now examine the properties of E_{Ave}.

9.4.2.3 Dependency of the Average Update Energy on the Group Size N, the Tree Degree α and the Deployment Region

In this section, we examine the dependency of the E_{Ave} on the group size N, the degree of the key distribution tree α, and the network deployment region. We do so by extracting an upper bound on E_{Ave} that does not depend on the distribution $p(M_i)$ of the member leaves/deletions, or the network topology.

E_{Ave} depends on the energy $\tilde{E}_{M_i}$ required for the deletion of each member M_i from the multicast group MG. Regardless of which member is deleted, the number of messages sent by the GC for updating keys after a member leave, is equal to $(\alpha \log_\alpha N - 1)$. These messages are routed to different subgroups SG of the multicast group MG (see Figure 9.3). However, the energy for sending a message from the GC to any subgroup SG of the multicast group MG, cannot

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

exceed the energy for sending a message to the whole group MG if the same routing tree is used in both cases. Though true for any routing tree, assuming that the optimal routing tree in total transmission power is used for message delivery,

$$E_{SG} \leq E^*_{MG}, \quad \forall\, SG \subseteq MG \tag{9.4}$$

where E^*_{MG} denotes the minimum energy consumed for sending a message from the GC to the whole multicast group, calculated according to the optimal routing tree, and E_{SG} denotes the energy consumed for sending a message from the GC to all members of the subgroup SG, calculated with the same routing tree. Note that E_{SG} need not necessarily be optimal. Using the inequality in (9.4), we can bound the energy $\tilde{E}_{M_i}$, for routing $(\alpha \log_\alpha N - 1)$ update messages to different subgroups of the multicast group MG by:

$$\tilde{E}_{M_i} \leq E^*_{MG}(\alpha \log_\alpha N - 1) \tag{9.5}$$

By combining (9.3) and (9.5), we can bound the average update energy E_{Ave} for a multicast group MG of size N and a key distribution tree of degree α:

$$\begin{aligned} E_{Ave} &= \sum_{i=1}^{N} p(M_i)\tilde{E}_{M_i} \\ &\leq \sum_{i=1}^{N} p(M_i)E^*_{MG}(\alpha \log_\alpha N - 1) \\ &\leq E^*_{MG}(\alpha \log_\alpha N - 1)\sum_{i=1}^{N} p(M_i) \\ &\leq E^*_{MG}\left(\alpha \log_\alpha N - 1\right) \end{aligned} \tag{9.6}$$

The bound in (9.6) has two different components. The first component is the minimum energy E^*_{MG} required for sending a message to the whole multicast group MG. E^*_{MG} depends on the wireless medium characteristics and the network topology/routing protocol that defines the routing tree. However, we can relax the network topology dependency by bounding E^*_{MG} using only the wireless medium characteristics and size of the deployment region.

Let γ_{max} be the maximum value of the attenuation factor for the heterogeneous medium where the network is deployed, and let d_{max} be the size of the deployment region, defined by the physical distance between the GC and the farthest member.* Assuming that omnidirectional antennas are used, the GC can broadcast a message to all members of the multicast group, just by transmitting to the farthest member located at d_{max}.[38] Under this routing strategy, the

* The size of the deployment region may also be defined as the maximum physical distance between any two nodes of the network. However, such a definition leads to a looser upper bound and is not considered.

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

transmission power of the GC for sending one message to all members of MG cannot exceed:

$$P_{max} \leq (d_{max})^{\gamma_{max}} \tag{9.7}$$

Hence, the energy expenditure, $E_{MG}^{broadcast}$ for broadcasting a message from the GC to all members of MG can be bounded as:

$$\begin{aligned} E_{MG}^{broadcast} &= P_{max} T_{trans} \\ &\leq (d_{max})^{\gamma_{max}} T_{trans} \end{aligned} \tag{9.8}$$

where T_{trans} is the duration of the transmission of one message, fixed by the size of the message and the transmission bit rate. However, E_{MG}^{*} is optimal for sending a message from the GC to *all* members of MG. Hence, the optimal energy E_{MG}^{*} corresponding to the minimum total power strategy, should not be higher than $E_{MG}^{broadcast}$. Therefore, E_{MG}^{*} in (9.6) can be bounded by:

$$\begin{aligned} E_{MG}^{*} &\leq E_{MG}^{broadcast} \\ &\leq (d_{max})^{\gamma_{max}} T_{trans} \end{aligned} \tag{9.9}$$

The second component of the bound in (9.6) is the number of update messages sent by the GC for deleting a member from the multicast group. Whereas the number of messages grows logarithmically with the group size N, and N is not a design parameter, we can calculate the tree degree α^* that minimizes the number of update messages:

$$\begin{aligned} \frac{d}{d\alpha}(\alpha \log_{\alpha} N - 1) &= 0 \\ \frac{\ln \alpha - 1}{\ln \alpha^2} &= 0 \\ \alpha^* &= e \end{aligned} \tag{9.10}$$

The degree of the tree has to be an integer number, and hence the lowest upper bound for E_{Ave} is achieved when $\alpha = 3$. The lowest upper bound for the average update energy, independent of the network topology and distribution of member leaves/deletions, is:

$$E_{Ave} \leq (3 \log_3 N)(d_{max})^{\gamma_{max}} T_{trans} \tag{9.11}$$

Note that if we optimize the tree degree α, to minimize the number of rekey messages when both joins and leaves are taken into account, and assuming that they occur equally likely, it can be shown that $\alpha^* = 4$.[42] Also, if we consider key trees using one-way functions as in Canetti et al.(1999)[23], the optimal tree degree α that minimizes the number of rekey messages is equal to $\alpha^* = 2$. The analysis presented in this section holds for both one-way function trees as in Canetti et al. (1999),[23] and joint consideration of joins and leaves as in

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

Wong et al. (2000),[42] with the upper bound in (9.11) adjusted according to the optimal tree degree in each case. In general, the optimal tree degree in (9.11) can be adjusted to correspond to any assumed model of joins and leaves, or any other key tree structure.

9.4.2.4 Impact of "Power Proximity" on the Key Management Overhead under the New Metric

In this section, we investigate the impact of the power proximity on the energy efficiency of the key distribution. By observing that in the homogeneous medium case (constant attenuation factor γ), power-proximity between two nodes is a monotonically increasing function of the physical distance between them, we show that we can perform energy-efficient key distribution by taking into account only the physical proximity of the nodes. We then show that when the medium is heterogeneous, location information alone is not sufficient for constructing an energy-efficient key-distribution scheme. In the case of a heterogeneous medium, we show that location has to be combined with path loss model information or power measurements in order to extract the power proximity between pairs of nodes, which allows us to construct energy-efficient key trees.

Network Deployed in a Homogeneous Medium (Constant Attenuation Factor) In a homogeneous medium, the transmission power for communication between nodes i, j is a monotonically increasing function of the distance $d_{i,j}$. Under the assumption that routing is optimally selected to minimize the total transmission power, spatially correlated nodes are connected in the routing tree or receive information through similar routing paths.[38] Intuitively, given the node location, members that are physically close should be grouped together and receive similar key updates to reduce the energy expenditure.

To illustrate the need for designing a location-aware key distribution, we consider the ad hoc network in Figure 9.4(a), which is deployed in a homogeneous medium. The routing tree shown in Figure 9.4(a) is optimal in total transmit power. In the key tree of Figure 9.4(c), denoted as Tree A, we randomly place the four members of the multicast group in the leaves of the key tree, independent of the network topology as in wired networks. Assume that key K_0 needs to be updated. On the first row of Table 9.2, for Tree A, we indicate the messages sent by the GC for the update of K_0 to the appropriate subgroups, and the corresponding energy expenditure. The energy is computed according to the optimal routing tree structure of Figure 9.4(a).

Assume now that the members are grouped according to their physical proximity. Then, M_1 is grouped with M_4, and M_2 with M_3, resulting in the location-aware key tree of Figure 9.4(d), denoted as Tree B. On the second row of Table 9.2, we indicate the messages sent by the GC to update K_0 to the appropriate subgroups, and the corresponding energy expenditure for Tree B. The energy

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

Table 9.2. *Messages Sent by the GC for the Update of K_0 and Associated Energy Expenditure for the Key-Distribution Trees of Figure 9.4(c), (d), (e). E^X_{rekey} Denotes the Energy Required for Updating K_0 in Key Tree X. $E_{A\to B}$ Denotes the Energy Required for Transmission of a Key from A to B*

Key Tree	Messages Sent by the GC	Energy Expenditure
Tree A	$GC \to \{M_1, M_3\} : \{K'_0\}_{K_{1.1}}$ $GC \to \{M_2, M_4\} : \{K'_0\}_{K_{1.2}}$	$E^A_{rekey} = E_{M_1\to M_4} + 2E_{GC\to M_2} + E_{M_2\to M_3}$
Tree B	$GC \to \{M_1, M_4\} : \{K'_0\}_{K_{1.1}}$ $GC \to \{M_2, M_3\} : \{K'_0\}_{K_{1.2}}$	$E^B_{rekey} = E_{GC\to M_1} + E_{M_1\to M_4} + E_{GC\to M_2} + E_{M_2\to M_3}$
Tree C	$GC \to \{M_1, M_2\} : \{K'_0\}_{K_{1.1}}$ $GC \to \{M_3, M_4\} : \{K'_0\}_{K_{1.2}}$	$E^C_{rekey} = E_{GC\to M_1} + E_{M_2\to M_3} + E_{GC\to M_2} + E_{M_3\to M_4}$

saved by performing a rekey operation with the location-aware key Tree B over the random key Tree A for the network of figure 9.4(a) is computed as:

$$E^A_{rekey} - E^B_{rekey} = E_{GC\to M_2} - E_{GC\to M_1} > 0 \tag{9.12}$$

where E^X_{rekey} denotes the energy required for updating K_0 in key Tree X and $E_{A\to B}$ denotes the energy required for transmission of a key from node A to

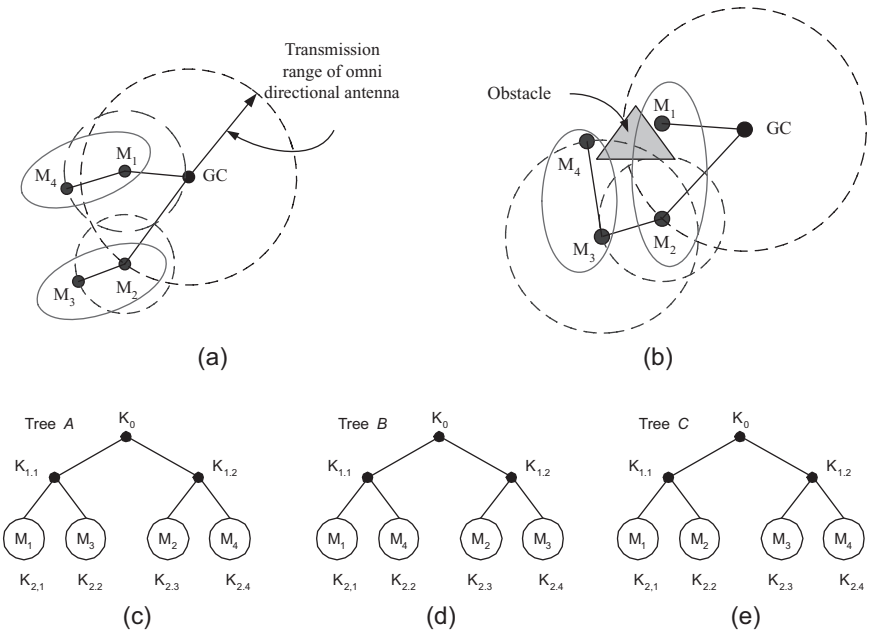

Figure 9.4. An ad hoc network and the corresponding routing tree with the minimum total transmission power, deployed in (a) a homogeneous medium and (b) a heterogeneous medium (c) A random key-distribution tree, Tree A. (d) A key-distribution tree based in physical proximity, Tree B, (e) A key-distribution tree based on "power proximity," Tree C.

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

node B. Non-negativity follows from the fact that $d_{GC,M_2} > d_{GC,M_1}$ and from the homogeneity of the medium (γ is constant). Hence, $P(d_{GC,M_2}) > P(d_{GC,M_1})$.

Network Deployed in a Heterogeneous Medium (Variable Attenuation Factor) We now consider the case of an ad hoc network deployed in a heterogeneous medium, where the attenuation factor γ varies significantly over different regions of the network. Under heterogeneous path loss, physical proximity of two nodes does not necessarily imply that the power needed for establishing a communication link is lower than the power needed for two nodes located farther apart. Thus, closely located nodes do not necessarily receive messages through similar routing paths. Hence, node location information alone is not sufficient for constructing an energy-efficient key tree.

To illustrate the preceding observation, we consider the ad hoc network shown in Figure 9.4(b), in which nodes have the same locations as in Figure 9.4(a). However, there exists a physical obstacle between nodes M_1 and M_4. Thus, the attenuation factor for signal transmission between M_1 and M_4 is significantly higher than the rest of the obstacle-free network regions. Therefore, the optimal routing tree in total transmission power connects M_4 to the network through M_3.

We now show that in an environment with variable path loss, we are able to construct an energy-efficient key tree by correlating nodes according to their power proximity rather than physical proximity. We may acquire such information either by using path loss information in addition to the node location, or by measuring the required transmission power for communication between pairs of nodes. Members that are closely located in terms of power are grouped together (placed adjacently to the key tree).

For the network in Figure 9.4(b), we construct the key distribution tree in Figure 9.4(e), denoted as Tree C. We place members adjacently to the key tree according to their power proximity. M_1 is grouped with M_2, and M_3 with M_4 in order to minimize the total communication power variance of clusters of two members. In the third row of Table 9.2, we indicate the messages sent by the GC for the update of K_0 to the appropriate subgroups and the corresponding energy expenditure for Tree C. The energy saved for performing a rekey operation by incorporating location as well as the path loss information instead of location alone is computed as the energy gain due to use of Tree C over Tree B:

$$E^B_{rekey} - E^C_{rekey} = E_{M_1 \to M_4} - E_{M_3 \to M_4} > 0. \tag{9.13}$$

Non-negativity follows from the observation that due to the obstacle between M_1 and M_4, $E_{M_1 \to M_4} > E_{M_3 \to M_4}$. Based on our analysis in Sections 9.4.2 and 9.4.2, we make the following conclusions:

Conclusion 1: When the medium is homogeneous, the transmission power $P(d_{i,j})$ is a monotonically increasing function of the distance $d_{i,j}$ between nodes

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

i and j, ($P(d_{i,j}) \propto d_{i,j}^{\gamma}$, γ constant). Hence, closely located nodes require less power for communication and therefore are connected in the routing tree that minimizes the total transmission power. By exploiting the physical proximity information, we can develop an energy-efficient key tree hierarchy.

Conclusion 2: In the case of a heterogeneous medium, no single function can map the distance to transmission power. Different functions with variable attenuation factor γ hold for different network regions. Hence, the use of physical proximity does not necessarily result in the minimum total power-routing tree. Instead, we use power proximity to create an energy-efficient key tree hierarchy.

Based on conclusions 1 and 2, we develop our key distribution algorithms for the homogeneous and heterogeneous cases.

9.4.3 Location-Aware Key Distribution for a Homogeneous Medium

In this section, we develop an energy-efficient key-distribution algorithm for the homogeneous medium, based on node location information. Note that updating the keys after a member deletion requires multicast transmissions to subgroups of various sizes (see Figure 9.3). For energy-efficient key distribution, we need to fully utilize the broadcast advantage when we distribute keys to subgroups.

In Section 9.4.2.4, we showed that placing closely located nodes adjacently on the key distribution tree results in significant savings in energy resources when the medium is homogeneous. In order to systematically construct a key tree hierarchy, we need to be able to cluster nodes based on the location information. The clustering of the nodes should allow us to form a hierarchy. Then we can translate the physical clustering of the nodes into a key tree hierarchy, thus obtaining an energy-efficient key distribution tree. Hence, the task of developing a location-aware key distribution scheme is reduced to the task of identifying (a) a location-based clustering mechanism, and (b) building a cluster hierarchy that utilizes the location-based clustering. We discuss both tasks in the following sections.

9.4.3.1 Location-Based Clustering for Energy-Efficient Key Distribution

For the homogeneous medium, we have set the constraint that the only information available to us is node location, without any explicit parametric model assumptions for our clustering. Hence, our clustering technique should be model-free while taking the location information into account. We also note that for the homogeneous case, physical proximity is a suitable metric because the attenuation factor γ is a constant. Hence, the Euclidean distance between the nodes is a natural metric for identifying and grouping neighbor nodes. Certainly some other distance metric, such as the Minkowsky metric,[45] can be used as well, but

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

the monotonicity of the power to the distance in the case of constant γ makes the Euclidean a very attractive one, since it leads to low-complexity algorithms.

Our effort is focused on finding a clustering technique that (a) requires only location information as an input, (b) identifies the physical network clusters with high success and, (c) generates clusters of equal size.

Problem Formulation for Location-Based Clustering

Let the coordinates of a node i be $x_i = \left(x_{i_1}, x_{i_2}\right)$. The squared Euclidean distance between two nodes i and i' is equal to:

$$d_{i,i'}^2 = \sum_{j=1}^{2} (x_{ij} - x_{i'j})^2 = \|x_i - x_{i'}\|^2 \tag{9.14}$$

If C denotes an assignment of the nodes of the network into α clusters, the dissimilarity function expressing the total intercluster dissimilarity $W(C)$ is:

$$W(C) = \sum_{k=1}^{\alpha} \sum_{C(i)=k} \|x_i - m_k\|^2 \tag{9.15}$$

where $C(i) = k$ denotes the assignment of the ith point to the kth cluster, and m_k is the mean (centroid) of cluster k. Intercluster dissimilarity refers to the dissimilarity between the nodes of the same cluster. We wish to find the optimal cluster configuration C^* that minimizes (9.15), subject to the constraint that the sizes of the resulting clusters are equal:

$$C^* = \arg\min_{C} \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - m_k\|^2, \quad \ni \quad |C(i)| = |C(j)|, \quad \forall i, j \tag{9.16}$$

Note that this formulation provides an optimal way to create α subclusters from one cluster. This location-based clustering has to be iteratively applied to generate the desired cluster hierarchy.

Solution Approach

If we relax the constraint $|C(i)| = |C(j)|, \quad \forall i, j$, in (9.16), and allow clusters of different sizes, the solution to the optimization problem in (9.16), can be efficiently approximated by K-means algorithm.[45] K-means uses squared Euclidean distance as a dissimilarity measure to cluster different objects. It also generates clusters by minimizing the total cluster variance (minimum square error approach). Note that K-means may result in a suboptimal local minimum solution depending on the initial selection of clusters, and hence, the best solution out of several random initial cluster assignments should be adopted.[45] However, K-means is easily implemented and hence, is an ideal solution for computationally limited devices. Algorithmic details on solving (9.16) without any constraint on the cluster size are given in Hastie et al. (2001).[45]

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

To satisfy the constraint posed in (9.16), we need a refinement algorithm (RA) that balances the cluster sizes while taking advantage of the low-complexity of K-means algorithm. According to (9.16), the RA should result in balanced clusters with the lowest total intercluster dissimilarity. In the binary tree case, given two clusters A, B with $|A| > |B|$, the refinement algorithm moves objects $i_1, i_2, \ldots, i_k \in A$, with $k = \lfloor \frac{|A|-|B|}{2} \rfloor$, from cluster A to cluster B, such that the intercluster dissimilarity after the refinement is minimally increased. We choose the objects $i_1, i_2, \ldots, i_k \in A$ such that:

$$i_j = \arg\min_{i \in A} \left[d^2_{i,m_B} - d^2_{i,m_A} \right], \quad j = 1 : \left\lfloor \frac{|A| - |B|}{2} \right\rfloor \tag{9.17}$$

where m_A and m_B refer to the centroids of clusters A and B, respectively. Note that in K-means, objects are assigned to the closest centroid, and hence, $[d^2_{i,m_B} > d^2_{i,m_A}], \forall i \in A$. By moving objects from A to B that increase $W(C)$ by the minimum possible amount, we achieve the optimal solution for the constrained optimization problem in (9.16) in the case of binary trees. *However, optimality is not guaranteed if more than two subclusters need to be balanced (d-ary tree).*

9.4.3.2 An Energy-Efficient Key-Distribution Scheme Based on Physical Proximity

We now develop an algorithm that maps the location-based clustering into a hierarchical key tree structure. Assume that we wish to construct a key tree of fixed degree α. Initially, the global cluster is divided into α subclusters using K-means. Considering that we want to construct a fixed-degree tree, every cluster *must* have equal number of members. Hence, we employ the RA algorithm to balance the cluster sizes. The RA leads to the construction of a balanced key tree when $N = \alpha^n$, $n \in \mathbb{Z}$, and allows us to construct a structure as close to the balanced as possible when $N \neq \alpha^n$. Each cluster is subsequently divided into α new ones, until clusters of at most α members are created (after $\log_\alpha N$ splits). Figure 9.5 presents the pseudo-code for our *Location-Aware Key Distribution Algorithm* (LocKeD) using K-means. We now describe the notational and algorithmic details of Figure 9.5.

Let $\mathcal{P}$ denote the set containing all the two-dimensional points (objects) corresponding to the location of the nodes. Let $C = \{C(1), C(2), \ldots, C(n)\}$ denote a partition of $\mathcal{P}$ into n subsets (clusters), that is, $\bigcup_i C(i) = \mathcal{P}$. Initially, all objects belong to the global cluster $\mathcal{P}$. The function *AssignKey()* assigns a key to every subset (cluster) of its argument set. For example, *AssignKey*($\mathcal{P}$) will assign the SEK to every member of the global cluster $\mathcal{P}$.

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

Location-Aware Key Distribution – LocKeD

$C = \{\mathcal{P}\}$
AssignKey(C)
index=1
while $index < \lceil \log_\alpha(N) \rceil$
  $C\_temp = \{\emptyset\}$
  $thres = \lceil \frac{N}{\alpha^{index}} \rceil$
  for $i = 1 : |C|$
    R=Kmeans$(C(i), \alpha)$
    R=Refine$(R, thres)$
    AssignKey(R)
    $C\_temp = C\_temp \bigcup R$
    index++
  end for
$C = C\_temp$
end while

(a)

Refinement Algorithm – RA

$C_{Low} = \{C(i) \in C : |C(i)| < thres\}$
$C_{High} = \{C(i) \in C : |C(i)| > thres\}$
repeat until $C_{High} = \emptyset$
  find $x^* \in A,\ A \in C_{High}$

$$x^* = \arg\min_{x \in A}[diss(x, m_B) - diss(x, m_A)],$$
$$\forall\, x \in A,\ \forall\, A \in C_{High},\ \forall\, B \in C_{Low}$$

  move x^* to cluster B

  $C_{Low} = \{C(i) \in C : |C(i)| < thres\}$
  $C_{High} = \{C(i) \in C : |C(i)| > thres\}$

end repeat

(b)

Figure 9.5. Pseudo-code for (a) the location-aware key-distribution algorithm (LocKeD) and (b) the Refinement Algorithm (RA). Repeated application of *Kmeans()* function followed by the Refinement Algorithm *Refine()* for balancing the clustering sizes, generates the cluster hierarchy. Function *AssignKey()* maps the cluster hierarchy into a tree hierarchy by assigning appropriate keys to cluster members.

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

The *index* variable counts the number of steps required until the termination of the algorithm. The *thres* variable holds the number of members each cluster ought to contain at level $l = index$ of the key tree construction. The root of the tree is at level $l = 0$. The $Kmeans(C(i), \alpha)$ function divides the set $C(i)$ into α clusters and returns the cluster configuration to variable R. The *Refine(R,thres)* function balances the sizes of clusters in R according to the *thres* variable. Then, *AssignKey()* is applied to assign different keys to every cluster in R. The process is repeated until $\lceil \log_\alpha N \rceil$ steps have been completed.

In terms of algorithmic complexity, the LocKeD algorithm iteratively applies K-means up to N times in the worst case (generation of a binary tree). K-means has algorithmic complexity of $O(N)$.[45] Hence, the complexity of the LocKeD is $O(N^2)$. We note that LocKeD requires only location information as input, assuming that the tree degree is fixed a priori.

Application of LocKeD on a Sample Network Deployed in a Homogeneous Medium

Consider the network in Figure 9.6(a), deployed in a homogeneous medium with an attenuation factor $\gamma = 2$. Assume that we wish to construct a location-aware key distribution tree of degree $\alpha = 2$ with nodes $\{2, 3, \ldots, 9\}$ being the members $\{M_2, M_3, \ldots, M_9\}$ of the multicast group, respectively. Initially, all members belong to the global cluster $\mathcal{P}$.

Note that the GC does not participate in the clustering. The hierarchical key tree is constructed by executing the following steps:

Step 1: Assign the SEK K_0 to every member of the global cluster $\mathcal{P}$.

Step 2: Create two clusters by splitting the global cluster. The two clusters that yield minimal total cluster dissimilarity are:

$$C_1 = \{M_2, M_3, M_4, M_6, M_8, M_9\}, C_2 = \{M_5, M_7\}.$$

Considering that we seek to construct a balanced key tree, apply the refinement algorithm to balance the clusters sizes. Move M_2 and M_6 to cluster C_2. Assign two different KEKs to members of clusters C_1 and C_2. Members of C_1 are assigned KEK $K_{1.1}$ and members of C_2 are assigned KEK $K_{1.2}$.

Step 3: Create clusters of two members by splitting the clusters of four members. The four created clusters are:

$$C_3 = \{M_2, M_6\},\ C_4 = \{M_3, M_4\}, C_5 = \{M_8, M_9\},\ C_6 = \{M_5, M_7\}.$$

Again, different KEKs are assigned to members of clusters C_3-C_6. Members of C_3 are assigned KEK $K_{2.1}$, members of C_4 are assigned KEK $K_{2.2}$, members of C_5 are assigned KEK $K_{2.3}$, and members of C_6 are assigned KEK $K_{2.4}$. At this point, we have completed the $\lceil \log_\alpha N \rceil$ steps required by LocKeD and the algorithm terminates.

The hierarchical key tree constructed using LocKeD is shown in Figure 9.6(b).

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

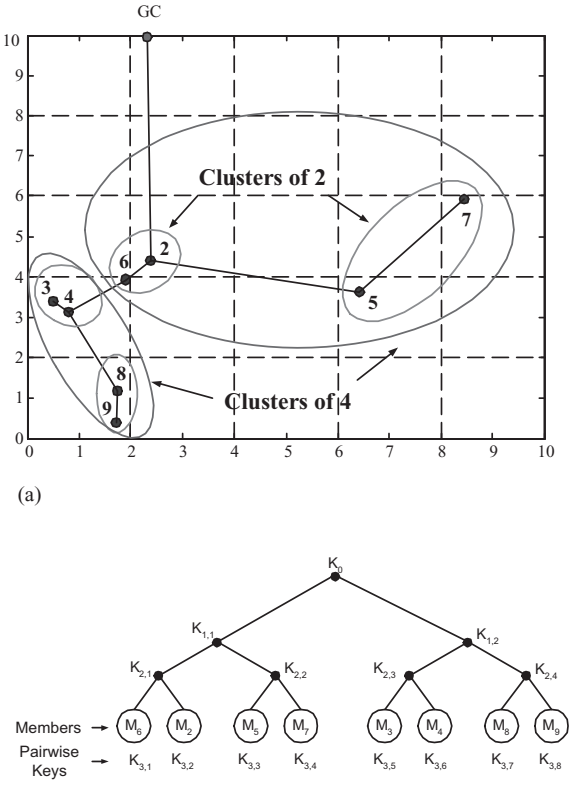

Figure 9.6. (a) An ad hoc network deployed in a homogeneous medium and the corresponding routing paths. Iterative application of the location-based clustering and the resulting cluster hierarchy. (b) The key-distribution tree resulting from the application of LocKeD.

When the medium is homogeneous, the key distribution algorithm discussed earlier makes use of the node location to securely and efficiently distribute keys to valid multicast group members. When the medium is heterogeneous, the node location information alone is not sufficient for energy-efficient clustering, and details are discussed in Lazos and Poovendran (2007) and Salido et al. (2007).[46,47] Location-based clustering algorithms were found to be power-efficient and secure when applied to obstacle-free open-space, suburban, and indoor environments.[46,47]

9.5 Using the Physical Layer to Enhance Security

The final component we will examine involves integrating the physical layer into the design of security protocols. The physical layer is responsible for the transmission and reception of signals between two or more entities. In the wireless context, the richness of the multipath environment associated with typical usage

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

scenarios (e.g., indoor or urban scenarios) implies that the physical characterization of the communication channel is a unique and hard-to-predict source of information shared between two communicators. More specifically, channel frequency responses decorrelate from one transmit-receive path to another if the paths are separated by the order of an RF wavelength or more.[48] The fact that pairwise radio propagation laws between two entities are unique and decorrelate quickly with distance can serve as the basis for establishing shared secrets. These shared secrets may be used as encryption keys for higher-layer applications or wireless system services that need confidentiality or secret keys.[49,50] Similarly, the wireless channel can enable wireless entities to authenticate other transmitters by tracking each other's ability to produce an appropriate received signal at the recipient.[51,55–58,60]

In this section, we shall focus on how authentication can be achieved at the lowest possible layer for a general wireless transmitter-and-receiver pair involving multiple transmit and receive antennas. For those interested in how the physical layer can be used for confidentiality, such as key establishment or secrecy dissemination, we refer the reader to Mathur et al. (2008).[50]

Prior work[52] on physical layer authentication has focused on single-antenna systems. However, with the ability to provide diversity gain and/or multiplexing gain, multiple-input multiple-output (MIMO) techniques will be widely deployed in future wireless networks – for example, IEEE 802.11n and WiMAX – to improve traffic capacity and link quality. The first caveat that must be understood when employing physical layer authentication is that channel-based authentication can only used to discriminate between different transmitters, and must be combined with a traditional handshake authentication process to completely identify an entity. In other words, identity is inherently a higher-layer function. Throughout this section, we assume that an entity's identity is obtained at the beginning of a transmission using traditional higher-layer authentication mechanisms. Consequently, the role that channel-based authentication plays is to ensure that all signals in both the handshake process and data transmission are actually from the same transmitter. Thus, this may be viewed as a cross-layer design approach to authentication.

9.5.1 System Model

As shown in Figure 9.7, Alice, Bob, and Eve are assumed to be located in spatially separated positions. Alice is the legal client with N_T antennas, initiating communication by sending signals to Bob. As the intended receiver, Bob is the legal access point (AP) with N_R antennas. Their nefarious adversary, Eve, will inject undesirable communications into the medium with N_E antennas, in the hopes of impersonating Alice.

We assume that Alice sends pilots from N_T antennas, and Bob uses these to estimate channel responses (we note that these pilots are typically used for

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

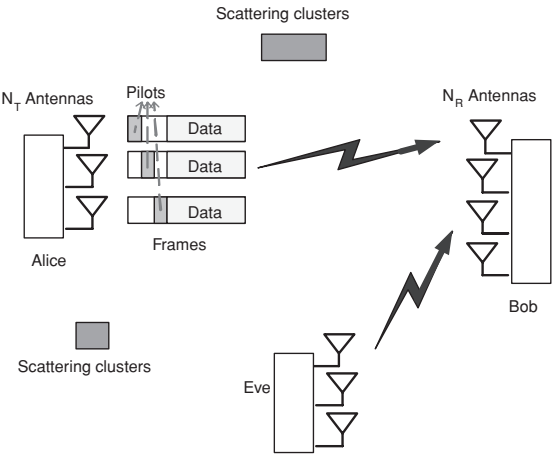

Figure 9.7. The adversarial multipath environment involving multiple scattering surfaces. The transmission from Alice with N_T antennas to Bob with N_R antennas experiences different multipath effects than the transmission by the adversary, Eve. Bob uses pilot symbols to estimate channel responses from the transmitters, and thus discriminate between Alice and Eve.

equalization purposes, and hence the security functions we describe can dual-use such physical layer information). In the authentication process, Bob tracks the channel responses to discriminate between legitimate signals from Alice and illegitimate signals from Eve.

A legal transmission from Alice to Bob in Figure 9.7 will involve a MIMO system with N_T transmit (Tx) antennas and N_R receive (Rx) antennas. Bob measures and stores channel frequency response samples at M tones, across an overall system bandwidth of W, where each subband has bandwidth $b\,(\leq W/M)$, and the center frequency of the system is f_0.

We consider channel frequency responses for two frames, which may or may not come from the same transmitter, and denote them by:

$$\mathbf{H}_i = \left[\underline{H}_i(1,1), \underline{H}_i(1,2), \ldots, \underline{H}_i(N_T, N_R)\right]^T, \quad i = 1, 2, \tag{9.18}$$

where $\underline{H}_i(j_t, j_r) = [H_{i,1}(j_t, j_r), \ldots, H_{i,M}(j_t, j_r)]^T$, $1 \leq j_t \leq N_T$, $1 \leq j_r \leq N_R$, and $H_{i,m}(j_t, j_r) = H_i(j_t, j_r, f_o + W(m/M - 0.5))$ is the channel response at the mth tone in the ith frame, connecting the j_tth Tx antenna and j_rth Rx antenna. The $N_T N_R M$ elements in $\mathbf{H}_i$ are independent and identically distributed. Considering the phase rotation and receiver thermal noise, one may model the estimated channel frequency response as $\hat{\mathbf{H}}_i = \mathbf{H}_i e^{j\phi_i} + \mathbf{N}_i$, where $\phi_i \in [0, 2\pi)$ denotes the unknown phase measurement rotation, and $\mathbf{N}_i$ is the receiver thermal noise vector with $N_T N_R M$ elements, which are independent and

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

identically distributed complex Gaussian random variables, $CN(0, \sigma^2)$, where σ^2 is the receiver noise power per tone.

MIMO-assisted channel-based authentication compares channel frequency responses at consecutive frames, and we should report spoofing if the channel responses from the same *claimed* user are significantly different in two frames. We note that we are assuming that the terminals are essentially stationary and that the channel does not vary significantly with time. Recent work[51] has examined the more general cases of time-variant channels.

We note that Eve can conduct an authentication attack only if she knows N_T and uses N_T transmit antennas. Following Kirkhoff's Principle for security analysis, we assume that Eve knows N_T. Assuming Bob obtains channel responses of $\hat{\mathbf{H}}_1$ and $\hat{\mathbf{H}}_2$, respectively, for two frames with the same identity, we build a simple hypothesis test for the purpose of transmitter discrimination. In the null hypothesis, $\mathcal{H}_0$, two estimates are from the same terminal, and thus the claimant is the legal user. Otherwise, Bob accepts the alternative hypothesis, $\mathcal{H}_1$, and claims that a spoofing attack has occurred.

The following test statistic is used to cope with unknown phase quantities ϕ_1 and ϕ_2:

$$L = \frac{1}{\sigma^2}||\hat{\mathbf{H}}_1 - \hat{\mathbf{H}}_2 e^{j\phi}||^2, \tag{9.19}$$

where

$$\phi = \arg\min_x ||\hat{\mathbf{H}}_1 - \hat{\mathbf{H}}_2 e^{jx}|| = Arg(\hat{\mathbf{H}}_1 \hat{\mathbf{H}}_2^H) \tag{9.20}$$

It can be shown under $\mathcal{H}_0$ that:

$$L_{\mathcal{H}_0} \approx \frac{1}{\sigma^2}||\mathbf{N}_1 - \mathbf{N}_2||^2 \sim \chi_S^2 \tag{9.21}$$

when the SNR is high, where $S = 2N_T N_R M$ degrees of freedom. Otherwise, when $\mathcal{H}_1$ is true, L is a noncentral Chi-square variable, given by:

$$L_{\mathcal{H}_1} \approx \frac{1}{\sigma^2}||\mathbf{H}_1 - \mathbf{H}_2 e^{j\phi} + \mathbf{N}_1 - \mathbf{N}_2||^2 \sim \chi_{S,\mu}^2 \tag{9.22}$$

where the noncentrality parameter, μ, is written as:

$$\mu = \frac{P_T}{P_N N_T}||\mathbf{H}_1 - \mathbf{H}_2 e^{jArg(\mathbf{H}_1 \mathbf{H}_2^H)}||^2 \tag{9.23}$$

For fixed P_T, the dimension of $\mathbf{H}_i$ is proportional to MN_R, and thus μ rises with both N_R and M. On the other hand, the impact of N_T is more complex, depending on the specific value of $\mathbf{H}_1$, $\mathbf{H}_2$, and P_T.

The rejection region of $\mathcal{H}_0$ is defined as $L \leq k$, where k is the test threshold, which is selected according to an appropriate performance target.

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

The performance of a physical layer authentication scheme should examine the "false alarm rate" for a given k as:

$$\alpha = Pr(L > k|\mathcal{H}_0) = 1 - F_{\chi^2_S}(k) \tag{9.24}$$

where $F_X(\cdot)$ is the CDF of the random variable X, as well as the "miss detection rate" for given k, which is given by:

$$\beta = Pr(L \leq k|\mathcal{H}_1) = F_{\chi^2_{S,\mu}}(k) \tag{9.25}$$

It can be seen that α rises with k, while β decreases with k, and further that the miss rate decreases with P_T.

The use of multiple antennas has a twofold impact: It improves security performance by increasing the frequency sample size from $2M$ to $2MN_TN_R$, and the use of multiple transmit antennas reduces the transmit power per antenna, leading to performance loss of some degree.

Note that the frequency sample size, $M \in [1, M_s]$, is selected for security purposes, where M_s ($\geq M$), the total number of subbands is determined by non-security issues such as data-decoding accuracy. The average transmit power per tone is determined by M_s, with $P_T = P_{total}/M_s$, where P_{total} is the total system transmit power. Hence, P_T is independent of any other parameters mentioned, and we assume constant P_T in the comparison of system configurations.

In wideband systems, b is fixed and the detection performance improves with W, since channel responses decorrelate more rapidly in space with higher system bandwidth. It can be seen that β increases with b, since the power of measurement noise is proportional to b.

To illustrate the performance of channel-based authentication, we present simulation results that were obtained using the WiSE ray-tracing tool.[59] WiSE was used to generate typical channel responses for different locations in a typical office building. A brief description of the scenario is now provided, and we refer the reader to Xiao et al. (2008a)[53] for more detailed descriptions. In the office building, we deployed Bob as an access point roughly in the middle of the building, and varied the locations for Alice and Eve throughout a region of the building. For every Alice and Eve location, we calculated β for a given α. We then aggregated the results over all Alice-Eve location pairs to understand the overall feasibility for an adversary to conduct spoofing in this environment.

In the simulations, we consider MIMO, single-input multiple-output (SIMO), multiple-input single-output (MISO), and single-input single-output (SISO) systems, with separation of two neighboring antennas of 3 cm (i.e., half wavelength), $\alpha = 0.01$, $f_0 = 5$ GHz, $N_F = 10$, $b = 0.25$ MHz, and $P_T \in \{0.1, 1, 10\}$ mW, if not specified otherwise. The per tone SNR ranges from -16.5 dB to 53.6 dB, with a median value of 16 dB, using transmit power per tone $P_T = 0.1$

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

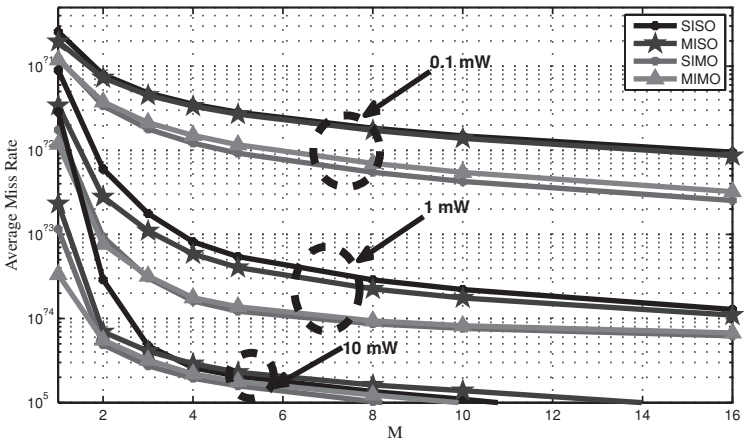

Figure 9.8. Average miss rate of spoofing detection in wideband systems, in SISO, 2×1 MISO, 1×2 SIMO, and 2×2 MIMO systems, respectively, with $\alpha = 0.01$, $M = 5$, $b = 0.25$ MHz, $W = 20$ MHz, and $P_T \in \{0.1, 1, 10\}$ mW.

mW, $b = 0.25$ MHz, and $N_T = N_R = 1$. Figure 9.8 shows that the average miss rate decreases with the frequency sample size, M, with a system bandwidth of $W = 20$ MHz, indicating that we should use all of the channel estimation data and set $M = M_s$. In addition, it can be seen that the security gain of MIMO decreases with M, when $P_T > 0.1$ mW. If using high power and small M (e.g., $M = 1$), the SISO system has accurate but insufficient channel response samples. Thus the additional dimensions of channel samples in MIMO systems allow for much better performance. On the contrary, if using high P_T and large M, the performance of SISO systems is too good to be significantly improved on.

We can also see that the security gain for a MIMO system over a SISO system slightly rises with M, when P_T is as low as 0.1 mW. This is because, when the channel estimation is not accurate due to low SNR, the system needs much more data to make a correct decision. Similarly, the impact of P_T on the MIMO security gain also depends on the value of M. The gain rises with P_T, under small M, whereas under large M, the security gain decreases with P_T.

We now summarize by examining how the above results on physical layer authentication should be interpreted in the context of future wireless networks. First, MIMO communication is becoming a dominant modality for wireless communication. Already technologies like 802.11n, which employ MIMO, are becoming prolific because of the communication rate and reliability improvements MIMO offers. It is possible to dual-use information associated with channel characterization, which is naturally obtained for the purposes of channel estimation for normal coding and decoding of signals, to provide a method

https://doi.org/10.1017/CBO9780511921117.010 Published online by Cambridge University Press

for distinguishing between transmitters. The typical false alarm and miss rates that one can expect from physical layer authentication, for example $\alpha \approx 0.1$ and $\beta \approx 10^{-4}$, suggests that physical layer authentication will not replace the strong authentication guarantees of classical cryptographic methods (such as message authentication codes and digital signatures). However, these values for α and β do imply that it is possible to use the physical layer as a lightweight authentication service, which can serve as an initial filter lightening the load on higher-layer authentication services,[54] or even as an anomaly-detection scheme that flags a network administrator of potential intrusions on an open network.

9.6 Concluding Remarks

In this chapter, we have explored several new approaches to integrating security into the design of a future Internet. The methods discussed all share the common feature that physical properties must be used in order to enhance security and, for the most part, do not employ traditional cryptographic mechanisms/protocols. These physical properties can vary from assuring the integrity of network devices themselves to making use of position information to check that transmissions are occurring where they should, or as a means to improve the efficiency of security functions (e.g., key management), or even to using the raw properties of the signals being transmitted and received so as to derive signatures that can discriminate between transmitters. The discussion in this chapter has primarily focused on wireless networks, and hence is targeted at the *edge* of the future Internet. To a large part, our discussion has focused on wireless networks because they represent the primary access technology that users will employ in the future (and hence security mechanisms at the edge are paramount to establishing a first line of defense for the broader network). Beyond this, though, wireless technologies are also the most rapidly evolving of communication technologies, where interfaces to all layers of the protocol stack are being made available to programmers for development, and thus such platforms also allow for an easy path to experiment with such nontraditional approaches to security. We would note, though, that there is no reason why the methods described in this chapter could not be employed on other networks, such as optical networks, with appropriate modifications.

Lastly, we would remark that the objective of this chapter is to highlight a complementary set of tools that can be used to provide additional security, and we emphasize that a starting point for securing any network should be a collection of properly designed security protocols at the various layers of the protocol stack, and which utilize cryptographic primitives that correctly interlock with each other across the various network layers. Unfortunately, this is generally a daunting task, and the methods outlined in this chapter can be viewed as tools that can assist when there are weaknesses inherent in the underlying security protocols.

References

[1] Wu, S. G. Z., and Raychaudhuri, D. 2006. Irma: Integrated Routing and MAC Scheduling in Multihop Wireless Mesh Networks. *Proceedings of IEEE WiMesh Workshop*.

[2] Google android smart phone. http://www.android.com/

[3] Apple iphone. http://www.apple.com/iphone/

[4] Apple iphone sdk. http://developer.apple.com/iphone/

[5] Google android sdk. http://code.google.com/android/

[6] Shi, E., Perrig, A., and Doorn, L. V. 2005. BIND: A Time-of-Use Attestation Service for Secure Distributed Systems. *Proceedings of IEEE Symposium on Security and Privacy*.

[7] Seshadri, A., Perrig, A., van Doorn, L., and Khosla, P. 2004. Swatt: Software-Based Attestation for Embedded Devices. *Proceedings of the IEEE Symposium on Security and Privacy*.

[8] Seshadri, A., Luk, M., Shi, E., Perrig, A., van Doorn, L., and Khosla, P. 2005. Pioneer: Verifying Integrity and Guaranteeing Execution of Code on Legacy Platforms. *Proceedings of ACM Symposium on Operating Systems Principles*, pages 1–16.

[9] Mitola, J., and Maguire, G. Q. 1999. Cognitive Radio: Making Software Radios More Personal. *Personal Communications, IEEE*, 6(4): 13–18.

[10] The GNU software radio. http://www.gnu.org/software/gnuradio/

[11] How to write a signal processing block for gnu radio. http://www.gnu.org/software/gnuradio/doc/howto-write-a-block.html

[12] Rice University WARP – wireless open-access research platform. http://warp.rice.edu

[13] WINLAB WIN2CR platform. http://www.winlab.rutgers.edu/docs/focus/WiNC2R.html

[14] Xu, W., Kamat, P., and Trappe, W. 2006. TRIESTE: A Trusted Radio Infrastructure for Enforcing Spectrum Etiquettes. *IEEE Workshop on Networking Technologies for Software Defined Radio (SDR) Networks*.

[15] Trusted computing group. http://www.trustedcomputinggroup.org/

[16] BBN Technologies. 2004. XG Policy Language Framework, Version 1.0. *XG Working Group Document*.

[17] Owl web ontology language guide. http://www.w3.org/TR/owl-guide/

[18] Bellardo, J., and Savage, S. 2003. 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions. *Proceedings of the USENIX Security Symposium*, pages 15–28.

[19] Bahl, P., and Padmanabhan, V. N. 2000. Radar: An In-building Rf-based User Location and Tracking System. *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 775–784.

[20] Youssef, M., Agrawal, A., and Shankar, A. U. 2003. WLAN Location Determination via Clustering and Probability Distributions. *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 143–150.

[21] Elnahrawy, E., Li, X., and Martin, R. P. 2004. The Limits of Localization Using Signal Strength: A Comparative Study. *Proceedings of the First IEEE International Conference on Sensor and Ad hoc Communcations and Networks (SECON 2004)*, pages 406–414.

[22] Yang, J., Chen, Y., and Trappe, W. 2009. Detecting Spoofing Attacks in Mobile Wireless Environments. *Proceedings of the Sixth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*.

[23] Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., and Pinkas, B. 1999. Multicast Security: A Taxonomy and Some Efficient Constructions. *Proc. of IEEE INFOCOM 99.*

[24] Sun, Y., Trappe, W., and Liu, R. 2002. An Efficient Key Management Scheme for Secure Wireless Multicast. *Proc. of IEEE Int. Conf. on Communication (ICC'02)*, vol. 2, pages 1236–1240.

[25] Sun, Y., Trappe, W., and Liu, R. 2003. Topology-Aware Key Management Schemes for Wireless Multicast. *Proc. of IEEE GLOBECOM*.

[26] Dommety, G., and Jain, R. 1996. Potential Networking Applications of Global Positioning Systems (GPS). *Technical report TR-24, the Ohio State University*.

[27] He, T., Huang, C., Blum, B. M., Stankovic, J. A., and Abdelzaher, T. 2003. Range-Free Localization Schemes in Large Scale Sensor Networks. *Proc. of the Ninth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*.

[28] Niculescu, D., and Nath, B. 2001. DV Based Positioning in Ad Hoc Networks. *Journal of Telecommunication Systems*.

[29] Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. 2000. The Cricket Location-Support System. *Proc. of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*.

[30] Savvides, A., Han, C., and Srivastava, M. 2001. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. *Proc. of the ACM Annual International Conference on Mobile Computing and Networking (MOBICOM 2001)*.

[31] Li, J., Jannotti, J., De Couto, D., Karger, D., and Morris, R. 2000. A Scalable Location Service for Geographic Ad-Hoc Routing. *Proc. of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*.

[32] Xue, Y., Li, B., and Nahrstedt, K. 2001. A Scalable Location Management Scheme in Mobile Ad-Hoc Networks. *Proc. of the IEEE Conference on Local Computer Networks (LCN 2001)*.

[33] Sastry, N., Shankar, U., and Wagner, D. 2003. Secure Location Verification. *Proc. of the ACM Workshop on Wireless Security (Wise 2003)*.

[34] Carman, D. W., Cirincione, G. H., and Matt, B. J. 2002. Energy-Efficient and Low-Latency Key Management for Sensor Networks. *Proc. of the 23rd Army Science Conference*.

[35] Stajano, F., and Anderson, R. 1999. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. *Security Protocols, 7th International Workshop*.

[36] Cagalj, M., Hubaux, J. P., and Enz, C. 2002. Minimum-Energy Broadcast In All Wireless Networks: NP-Completeness and Distribution Issues. *Proc. of the 8th ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2002)*.

[37] Liang, W. 2002. Constructing Minimum-Energy Broadcast Trees in Wireless Ad Hoc Networks. *Proc. of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002)*.

[38] Wieselthier, J. E., Nguyen, G. D., and Ephremides, A. 2000. On the Construction of Energy Efficient Broadcast and Multicast Trees in Wireless Networks. *Proc. of IEEE INFOCOM 2000*, pages 586–594.

[39] Lee, S., Su, W., Hsu, J., Gerla, M., and Bagrodia, R. 2000. A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols. *Proc. of the IEEE Conference on Computer Communications (INFOCOM 2000)*, pages 565–574.

[40] Rappaport, T. 1996. *Wireless Communications: Principles & Practice*, Prentice Hall.

[41] Raghunathan, V., Schurgers, C., Park, S., and Srivastava, M. B. 2002. Energy-Aware Wireless Microsensor Networks. *IEEE Signal Processing Magazine,* 19(2), 40–50.

[42] Wong, C. K., Gouda, M., and Lam, S. 2000. Secure Group Communications Using Key Graphs. *IEEE/ACM Trans. On Networking* 8(1), 16–31.

[43] Mittra, S. 1997. Iolus: A Framework for Scalable Secure Multicasting. *Proc. of SIGCOMM*.

[44] Wallner, D. M., Harder, E. C., and Agee, R. C. 1998. Key Management for Multicast: Issues and Architectures. *INTERNET DRAFT*.

[45] Hastie, T., Tisbshirani, R., and Friedman, J. 2001. *The Elements of Statistical Learning, Data Mining, Inference and Prediction*, Springer Series in Statistics, New York.

[46] Lazos, L., Poovendran, R. 2007. Power Proximity Based Key Management for Secure Multicast in Ad Hoc Networks. *Wireless Networks*, 13(1), 127–148.

[47] Salido, J., Lazos, L., and Poovendran, R. 2007. Energy and Bandwidth-Efficient Key Distribution in Wireless Ad Hoc Networks: A Cross-Layer Approach. *IEEE/ACM Transactions on Networking*, 15(6), 1527–1540.

[48] Jakes, W. C. 1994. *Microwave Mobile Communications*. Wiley-IEEE Press.

[49] Li, Z., Trappe, W., and Yates, R. 2007. Secret Communication Via Multi-Antenna Transmission. *Information Sciences and Systems, 2007. CISS '07. 41st Annual Conference on*, pages 905–910.

[50] Mathur, S., Trappe, W., Mandayam, N., Ye, C., and Reznik, A. 2008. Radio-Telepathy: Extracting a Secret Key from an Unauthenticated Wireless Channel. *MobiCom '08: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, pages 128–139.

[51] Xiao, L., Greenstein, L., Mandayam, N., and Trappe, W. 2008. Using the Physical Layer for Wireless Authentication in Time-Variant Channels. *IEEE Trans. on Communications*, pages 2571–2579.

[52] Xiao, L., Greenstein, L., Mandayam, N., and Trappe, W. 2007. Fingerprints in the Ether: Using the Physical Layer for Wireless Authentication. *Proc. IEEE International Conference on Communications (ICC)*, pages 4646–4651.

[53] Xiao, L., Greenstein, L., Mandayam, N., and Trappe, W. 2008. MIMO-Assisted Channel-Based Authentication in Wireless Networks. *Proc. Conference on Information Sciences and Systems (CISS)*, pages 642–646.

[54] Xiao, L., Greenstein, L., Mandayam, N., and Trappe, W. 2008. A Physical-Layer Technique to Enhance Authentication for Mobile Terminals. *Proc. IEEE International Conference on Communications (ICC)*.

[55] Xiao, L., Greenstein, L. J., Mandayam, N. B., and Trappe, W. 2008. A Physical-Layer Technique to Enhance Authentication for Mobile Terminals. *ICC '08. IEEE International Conference on Communications*, pages 1520–1524.

[56] Xiao, L., Greenstein, L. J., Mandayam, N. B., and Trappe, W. 2008. Mimo-Assisted Channel-Based Authentication in Wireless Networks. *Proceedings of the International Conference on Information Sciences and Systems (CISS)*.

[57] Xiao, L., Greenstein, L. J., Mandayam, N. B., and Trappe, W. 2009. Fingerprints in the Ether: Using the Physical Layer for Wireless Authentication. *CoRR*, abs/0907.4877.

[58] Xiao, L., Greenstein, L. J., Mandayam, N. B., and Trappe, W. 2009. Using the Physical Layer for Wireless Authentication in Time-Variant Channels. *CoRR*, abs/0907.4919.

[59] Li, Z., Xu, W., Miller, R., and Trappe, W. 2006. Securing Wireless Systems Via Lower Layer Enforcements. *Proceedings of the 2006 ACM Workshop on Wireless Security*, pages 33–42.

[60] Yang, J., Chen, Y., and Trappe, W. 2008. Detecting Sybil Attacks in Wireless and Sensor Networks Using Cluster Analysis. *Proceedings of the Fourth IEEE International Workshop on Wireless and Sensor Networks Security*.