Chapter 16

Wireless Sensor Network Security

Harsh Kupwade Patil
Southern Methodist University

Thomas M. Chen
Swansea University

1. INTRODUCTION TO THE WIRELESS SENSOR NETWORK (WSN)

In recent times, advances in microelectronic mechanical systems (MEMS) and wireless communication technologies have led to the proliferation of wireless sensor networks (WSNs). A WSN can be broadly described as a network of nodes that makes a collaborative effort in sensing certain specified data around its periphery and thereby controls the surrounding environment. A typical sensor network consists of a large number of low-cost, low-powered sensor nodes that are deployable in harsh operating environments. Because of their varied applications in civilian and military sectors, WSNs have gained a lot of popularity in the past decade. For instance, applications can include habitat monitoring, air and water quality management, hazard and disaster monitoring, health care, remote sensing, smart homes, and so on. Figure 16.1 depicts a usual wireless sensor network in which sensor nodes are distributed in an ad-hoc, decentralized fashion. Usually, WSNs are connected to a legacy network (IP network or 3 G network) using one or more sink nodes or base stations. Furthermore, routing in WSN is typically carried in a hop-by-hop fashion.

In general, WSN protocols should be designed to minimize energy consumption and preserve the life of the network. Information gathering in WSN is done by asking for information regarding a specific attribute of the phenomena or by asking for statistics about a specific area of the sensor field. This requires a protocol that can handle requests for a specific type of information, which includes datacentric routing and data aggregation. The last important characteristic of wireless sensor networks is that the position of the nodes may not be engineered or predetermined, and therefore, must provide data routes that are self-organizing.

Although WSNs have gained a lot of popularity, they present some serious limitations when one is implementing security. WSNs present extreme resource limitations in available storage (memory) space, computing, battery life, and bandwidth. Hence, sensor networks present major challenges for integrating traditional security techniques in such resource-constraint networks. In addition, the ad-hoc, decentralized nature of WSNs would pose even greater challenges in applying conventional security mechanisms. Hence, researchers face the challenge of taking all these constraints in consideration while providing adequate security to such sensor networks.

WSN Architecture and Protocol Stack

Most of the traditional networks (for example, IP networks) are built on the Open System Interconnection (OSI) model. However, WSNs operate in a resource-constrained environment and therefore deviate from the traditional OSI model. A WSN stack usually consists of six layers: an application layer, middleware, transport, network, data link, and physical layer. In addition to these six layers that are mapped to each sensor node, there are three more planes that span across the entire sensor network and have more visibility to address issues such as mobility, power, and task management (see Figure 16.2).

Application Layer

The application layer aims to create an abstraction of the main functions of the sensing application, thereby making the lower software and hardware levels transparent to the end user. The application layer involves several processes running simultaneously and handles user requests relating to data aggregation, location finding, sleep/awake cycle control, time synchronization, authentication, encryption, key distribution, and other security measures. It also defines the order and format of message exchange between the two communication parties.

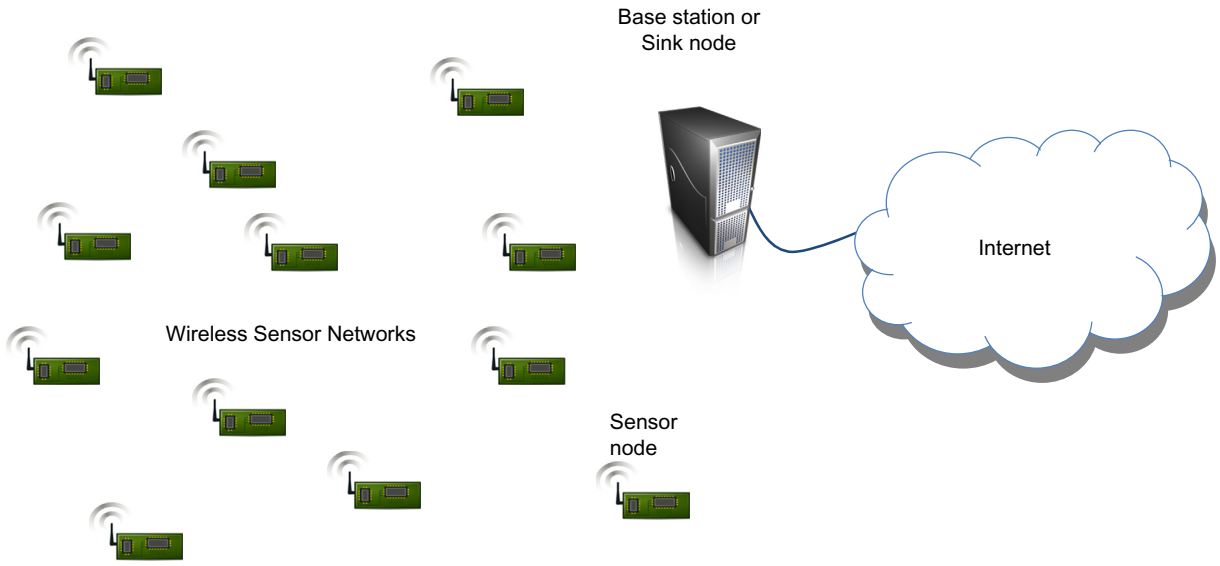

FIGURE 16.1 Wireless sensor network.

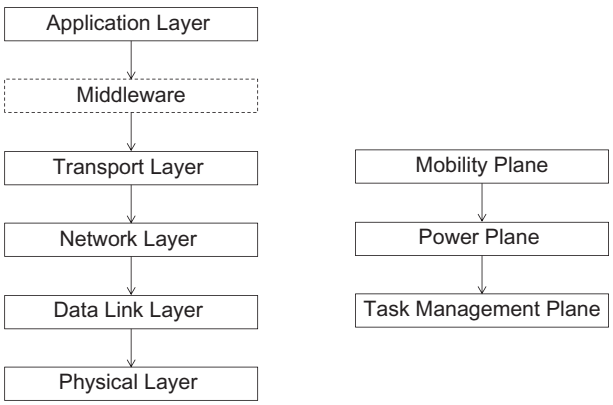

FIGURE 16.2 WSN protocol stack.

Middleware

The middle layer provides an application programming interface (API) for applications existing in the upper layers. It also may involve complex functionalities such as resource sharing and task management.

Transport Layer

The transport layer is responsible for flow and congestion control. It also performs error control to detect corrupted frames that arrive from lower layers. Due to the severe operating environment and reduced transmission power, it is difficult to achieve high end-to-end link reliability compared to traditional wireless networks. In addition, the transport layer performs fragmentation of sender data and reassembly of received data frames.

Network Layer

The network layer's primary goals are to perform routing operations and self-configuration. It is responsible for link failures and provides regular updates to neighboring nodes. However, assuring network connectivity at all times is a major challenge due to dynamically changing network topology. The routing protocols in WSN are very different from traditional routing protocols because of the need to optimize network life by performing intelligent routing.

Data Link Layer

The data link layer is an interface between the network and physical layer. It is further subdivided into two modules: Medium Access Control (MAC) and Logical Link Control (LLC). The MAC module plays a critical role in conserving network life by efficiently allocating medium access to the contending nodes. The LLC is on top of the MAC layer and is responsible for cyclic redundancy check (CRC), sequencing information, and addition of appropriate source and destination information. The data link layer is also responsible for the multiplexing of data streams and data frame detection. So, with the preceding in mind: first create a network infrastructure, which includes establishing communication links between possibly thousands of nodes, and provides the network self-organizing capabilities. Second, the data link layer can fairly and efficiently share communication resources between all the nodes.

Physical Layer

The physical layer is responsible for converting digital bits into analog symbols and vice versa. It involves

modulation and demodulation, frequency selection, power control, and symbol synchronization. WSNs usually operate in frequencies ranging from 915 MHz to 2.4 GHz. It is recommended using a lower-frequency band, as there is higher attenuation when operating in higher-frequency bands. However, with the limited availability of the bandwidth in the lower frequencies, the WSN is forced to operate at higher frequencies. The environment in which sensors are operating plays a major role in signal attenuation. Thus, sensors placed on the ground or floating on water experience greater attenuation and consequently require higher transmit power. The choice of modulation scheme is one of the prime factors in deciding the transmit power. The modulation scheme decides the bit error rate (BER), spectrum efficiency, and number of bits per symbol. For example, an M-ary modulation scheme is able to transmit more bits per symbol than other binary modulation schemes such as Phase Shift Keying (PSK). However, M-ary schemes result in higher BERs and require more transmit power than the binary modulation schemes. Hence binary modulation schemes are more applicable to WSN.

Mobility Plane

Sensor nodes can be fixed on moving objects such as animals, vehicles, and people, which will lead to a dynamic topology. In the event of some mobility by sensor nodes, the mobility in collaboration with the network layer is responsible for maintaining the list of active neighboring nodes. It is also responsible for interacting periodically with the mobility planes of other neighboring nodes, so that it can create and maintain a table of active, power-efficient routes.

Power Plane

The power plane focuses on the awareness of power at each horizontal and vertical layer. It is responsible for shutting off the sensors if they are not participating in any routing decisions or simply if the sensing activity is complete. The power planes of each node work collectively on deciding efficient routes to sink nodes and maintain the sleep/awake cycles of sensor nodes.

Task Management Plane

The task management plane is responsible for achieving a common goal. The goal is met by taking the properties of each layer and across each layer in a power-aware manner.

Vulnerabilities and Attacks on WSN

A taxonomy allows organizations to reason about attacks at a level higher than a simple list of vulnerabilities (see Figure 16.3). It provides a classification system that ideally suggests ways to mitigate attacks by prevention, detection, and recovery.

In general, attacks can be divided into active and passive attacks:

Passive Attack

In this type of attack, the attacker is able to intercept and monitor data between communicating nodes, but does not tamper or modify packets for fear of raising suspicion of malicious activity among the nodes. For example, in traffic analysis, the attacker may not be able to decode encrypted data, but can find useful information by analyzing headers of packets, their sizes, and the frequency of transmission. In WSN, reconnaissance can also be performed to understand information exchange between communicating nodes, particularly at data aggregation points. Furthermore, routing information can be exploited using traffic analysis.

Active Attack

In this type of attack, the attacker actively participates in all forms of communication (control and data) and may modify, delete, reorder, and replay messages or even send spoofed illicit messages to nodes in the network. Some other active attacks include node capturing, tampering with routing information, and resource exhaustion attacks. Peculiar to WSN, the attacker can modify the

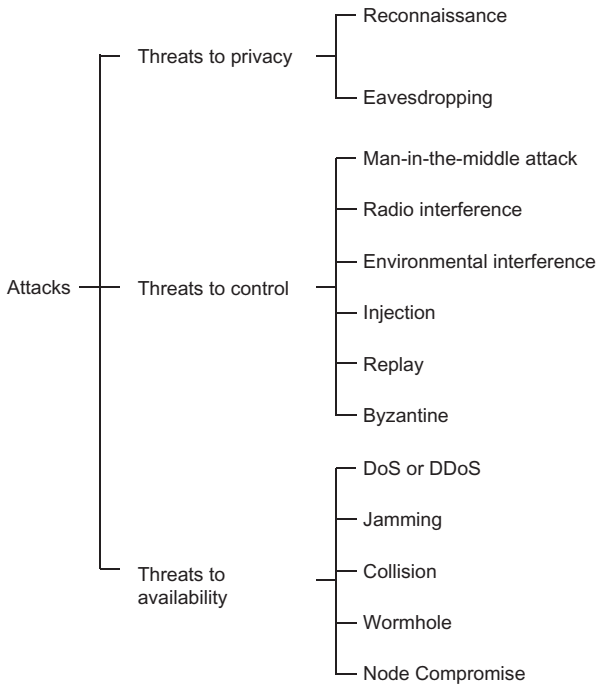

FIGURE 16.3 Taxonomy of attacks for WSN.

environment surrounding sensors, which could affect the sensed phenomena.

2. THREATS TO PRIVACY

In WSN, threats to privacy can be further classified into reconnaissance and eavesdropping.

Reconnaissance

Reconnaissance refers to intelligence gathering or probing to access the vulnerabilities in a network in order to launch a full-scale attack later. Reconnaissance attacks can be further classified into active and passive. Passive reconnaissance attacks include the collection of network information through indirect or direct methods, but without probing the target; active reconnaissance attacks involve the process of gathering traffic with the intention of eliciting responses from the target.

Eavesdropping

Eavesdropping is the act of listening secretly to a private conversation. However, in the paradigm of WSN, eavesdropping is an operation to learn the "aggregate data" that is being collected by the entire network. Hence, eavesdropping between two specific sensor nodes may not help the attacker in thoroughly understanding the entire network. It can be further classified into *active* and *passive* eavesdropping:

- *Active eavesdropping:* In this case, the adversary actively sends queries to other nodes in an attempt to goad them to respond to his queries, and in exchange will be able to comprehend the precise task assigned to the nodes in the network. Usually, the attacker launches a "man-in-the-middle attack" (discussed below) to infiltrate the network and enforce himself on the active path.
- *Passive eavesdropping:* The attacker inserts him- or herself into the active path, unbeknownst to other nodes in the network. He or she then passively listens to all traffic sent over the broadcast medium. It may be difficult to detect a passive eavesdropping attack, as the attacker may be operating in a stealth mode.

Threats to Control

The nodes in the network are unaware that the entire flow control is being handled by the attacker.

Man-in-the-Middle Attack

The man-in-the-middle attack is one of the classical attacks that can be executed in a WSN environment. In this type of attack, the attacker intrudes into the network and attempts to establish an independent connection between a set of nodes and the sink node. He can be in either a passive or an active state. In a passive state, he simply relays every message among the nodes with the intention of performing an eavesdropping attack. In an active state, he can tamper with the intercepted data in an effort to break authentication. In addition, the attack can be executed at the physical, data link, network, and application layers [1].

Radio Interference

With the increase in the number of wireless technologies using the same open spectrum band (2.4 GHz, 5 GHz, or 900 MHz), there is bound to be radio interference. For example, in a dense urban environment, where cordless phones share the same spectrum, radio interference can cause a sharp degradation of individual node performance. Similar problems can be projected for sensor networks with the increase in sensor nodes per network. The result of such interference could lead to change in the information bits transmitted over the wireless medium, thereby making the bits unintelligible and ultimately being dropped by the receiver [2]. Hence, radio interference could lead to a denial-of-service attack. The worst-case scenario in radio interference is jamming.

Injection Attack

After the attacker has clandestinely intruded into the WSN network, he may impersonate a few of the sensor nodes (or even sink nodes) and may inject malicious data into the network. The malicious data might be false advertisement of neighbor-node information to other nodes, leading to impersonation of sink nodes and aggregation of all data.

Replay Attack

A replay attack is a common attack in WSN, whereby an attacker is able to intercept user data and retransmit user data at a later time. This attack is particularly useful in breaking weak authentication schemes, which do not consider the time stamp when authenticating nodes. This attack is also useful during shared key-distribution processes.

Byzantine Attack

In a Byzantine attack, the outside adversary is able to take full control of a subset of authenticated nodes that can be further used to attack the network from inside. Such attacks by malicious behavior are known as Byzantine attacks. Some examples of Byzantine attacks

are black holes, flood rushing, wormholes, and overlay network wormholes:

- *Black-hole attack*: In this type of attack, the attacker drops packets selectively, or all control and data packets that are routed through him. Therefore, any packet routed through this intermediate malicious node will suffer from partial or total data loss.
- *Flood rushing attack*: This type of attack is common to wireless networks and exploits the flood duplicate suppression technique. In this attack, the attacker attempts to overthrow the existing routing path by sending a flood of packets through an alternate route, which will result in discarding the legitimate route and adopting the adversarial route. Usual authentication schemes cannot prevent this attack, as the adversaries are authenticated nodes.
- *Wormhole attack:* In this type of attack, two conniving sensor nodes, or laptops, tunnel control and data packets between each other, with the intention of creating a shortcut in the WSN. Such a low-latency tunnel between the two conniving nodes will likely increase the probability of it being selected as an active path. This type of attack is very closely related to the sinkhole attack, because one of the conniving nodes could falsely advertise to be the sink node and thereby attract more traffic than usual. One of the main differences between a Byzantine wormhole and a traditional wormhole is that in a Byzantine wormhole, the tunnel exists between two compromised nodes, while in a traditional wormhole, two legitimate nodes are tricked into believing that a secure tunnel exists between them.
- *Byzantine overlay network wormhole attack:* This type of attack is a variant of the wormhole attack and occurs when the wormhole attack is extended to multiple sensor nodes; resulting in an overlay of compromised nodes. It provides a false illusion, to honest nodes, that they are surrounded by legitimate nodes, resulting in frequent reuse of the adversarial path.

Sybil Attack

The Sybil attack was first introduced by John R. Douceur while studying security in peer-to-peer networks [3], and later Karlof and Wagner showed that this type of attack poses a serious threat to routing mechanisms in WSN [4]. Sybil is an impersonation attack in which a malicious node masquerades as a set of nodes by claiming false identities, or generating new identities in the worst case [5]. Such attacks can be easily executed in a WSN environment because the nodes are invariably deployed in an unstructured and distributed environment, and communicate via radio transmission. They are especially detrimental in applications such as data aggregation, voting systems, reputation evaluation, and geographic routing. By using a Sybil attack in location-aware routing, it is possible to be in multiple locations at the same time.

Sinkhole Attack

In a sinkhole attack, the adversary impersonates a sink node and attracts the whole of traffic to a node or a set of nodes. Similar to a black-hole attack, the attacker takes control of a few compromised nodes and advertises false routing information to its neighbors, thereby luring all traffic to him.

Threats to Availability

Due to threats to the WSN, some portion of the network or some of the functionalities or services provided by the network could be damaged and unavailable to the participants of the network. For instance, some sensors could die earlier than their expected lifetimes. Thus, availability service ensures that the necessary functionalities or the services provided by the WSN are always carried out, even in the case of attacks.

Denial of Service (DoS) or DDoS

A denial-of-service attack occurs when an attacker floods the victim with bogus or spoofed packets with the intent of lowering the victim's response rate. In the worst-case scenario, it makes the victim totally unresponsive. For instance, in a WSN environment where nodes have limited computational capacity, a DoS attack from a resource-abundant adversary can overwhelm the nodes by flooding packets, which will exhaust communication bandwidth, memory, and processing power. From an attacker's point of view, this attack is also useful in wireless networks where nodes are required to deliver time-critical data. Jamming the wireless links can also lead to a DoS attack. An extension of a DoS attack is a distributed DoS attack, where an attacker takes control of a few nodes in the network, leading to a distributed flood attack against the victim.

HELLO Flood Attack

One of the common techniques for discovering neighbors is to send HELLO packets. If a node receives a HELLO packet, it indicates that it is within the range of communication. However, a laptop-class adversary could easily send HELLO packets with sufficient power to convince the sensor nodes that it is in proximity of communication and may be a potential neighbor. The adversary could also impersonate a sink node or a cluster node.

Jamming

Jamming is one of the most lethal types of attacks in WSN and is a direct way to compromise the entire wireless network. In this type of attack, the attacker jams a spectrum band with a powerful transmitter and prevents any member of the network in the affected area from transmitting or receiving any packet. Jamming attacks can be divided into constant jamming and sporadic jamming. Sporadic jamming can be very effective at times when a change in one bit of a data frame will force the receiver to drop it. In this kind of attack, it is difficult for the victim to identify whether his band is being jammed intentionally or due to channel interference; his immediate reaction is usually to increase his transmitting power, thereby depleting resources at a faster rate. Jamming attacks target the physical and MAC layers. Four types of jamming attacks (random, reactive, deceptive, and constant) would result in DoS attacks [6]. Xu et al. conclude that intrusion detection schemes can be very complex with reference to differentiating malicious attacks from link impairment.

Collision

Collision attacks target the MAC layer to create costly exponential backoff. Whenever collision occurs, the nodes should retransmit packets affected by collision, thus leading to multiple retransmissions. The amount of energy expended by the attacker is much less than the energy expended (battery exhaustion) by the sensor nodes. Collision attacks can be categorized under resource exhaustion attacks.

Node Compromise

Node compromise is one of the most common and detrimental attacks in WSN. As sensors can be deployed in harsh environments such as a battlefield, ocean bed, or the edge of an active volcano, they are easily susceptible to capture by a foreign agent. In the case of a battlefield scenario, the enemy could make an effort to dig into nodes with the intention of extracting useful data (extracting private keys in sensor nodes). Furthermore, it could be reprogrammed and launched into a battlefield to operate on behalf of the enemy.

Attacks Specific to WSN

Wireless sensor networks are vulnerable to eavesdropping problems as the data transmission highly depends on the assumption that the receiving node faithfully receives and forwards the same transmitted packet containing specified parameters. But during peer-to-peer communication the parameters may be spoofed, replaced, altered, repeated, or even diminished by the single frequency or intentional intruders who can easily analyze the traffic flow and fabricate new parameters containing wrong information and transmit them to the sink nodes.

Attacks on Beaconing Protocol

A beaconing protocol uses a breadth-first spanning tree algorithm to broadcast routing updates. The sink node periodically broadcasts updated routing information to its immediate neighboring nodes. These neighboring nodes then rebroadcast this information to their immediate neighbors, and the process continues recursively. During this process, each intermediate node makes a note of its parent node (the parent node is the first node that was able to make contact with its subordinate node and relay the routing information). When all the active nodes are operational, they should send all the sensed data to their parent node. However, this protocol is vulnerable to many attacks. For example, a simple impersonation attack, leading to a sinkhole attack, can totally compromise the entire network [4,7].

Authentication can be used to prevent such impersonation attacks, but it does not prevent a laptop-class adversary from launching a selective forwarding attack, an eavesdropping attack, or a black-hole attack. The attacker creates a wormhole between two conniving laptop-class adversaries. The two laptops are placed near the sink node and the targeted area, respectively. The laptop near the sink node attracts its entire neighbor's traffic and simply tunnels these authenticated messages to its colluder. The laptop attacker, close to the sink node, plays a passive role in forwarding these messages. Due to his furtive nature, it is difficult for his neighbors to detect whether he is malicious. Once the authenticated messages reach the remote laptop adversary, he could launch a black-hole attack or a selective forwarding attack.

Let us consider a situation in which digital signatures are being used for authentication and, while the routing updates are in progress, the sink node's private key is leaked. As soon as the sink node realizes that its private key is being compromised, it immediately broadcasts a new public key. All the nodes in close proximity to the sink node will update their local copy of the sink node's public key. The laptop close to the sink node will perform the same operation and convey this information to its colluding laptop. The remote laptop can now easily impersonate the sink node and launch a sinkhole attack. In addition, she can further create routing loops, which is a resource-exhaustion attack.

Attacks on Geographic- and Energy-Aware Routing (GEAR)

GEAR proposes a location- and energy-aware, recursive routing algorithm to address the problem of uneven energy consumption in routing in WSN. In GEAR, every node gauges the energy levels of its neighbors along with the distance from the target before making a routing decision. In such situations, a laptop-class attacker can advertise that he has larger energy levels than his neighboring node and attract all traffic to him. Thenceforth, he can execute a Sybil, black-hole, or selective forwarding attack.

As attacks on WSN become more sophisticated, the demand for new security solutions is continually increasing. Hence, an array of new security schemes have been designed and implemented in the past decade [8,9]. Most of these schemes have been designed to provide solutions on a layer-by-layer basis rather than on a per-attack basis; in doing so, they have left a gap between layers that may lead to cross-layer attacks.

In general, any security suite should ensure authentication, integrity, confidentiality, availability, access control, and nonrepudiation (see checklist: An Agenda for Action When Implementing a Security Suite). In addition, physical safety is absolutely necessary to avoid tampering or destruction of nodes.

Security in WSN Using a Layered Approach

Most researchers have come up with a security solution to WSN based on a layered approach. However, a layered approach has noticeable flaws such as redundant security or inflexible security solutions.

Security Measures in the Physical Layer

To prevent radio interference or jamming, the two common techniques used are frequency-hopping spread spectrum (FHSS) and direct-sequence spread spectrum (DSSS). In FHSS, the signal is modulated at frequencies such that it hops from one frequency to another in a random fashion at a fixed time interval. The transmitter and the corresponding receiver hop between frequencies using the same pseudorandom code for modulation and demodulation. If an eavesdropper intercepts a FHSS signal, unless she has prior knowledge of the spreading signal code, she will not be able to demodulate the signal. Furthermore, spreading the signal across multiple frequencies will considerably reduce interference.

In DSSS, a spreading code is used to map each data bit in the original signal to multiple bits in the transmitted signal. The pseudorandom code (spreading code) spreads the input data across a wider frequency range compared

An Agenda for Action when Implementing a Security Suite

A construction of tamper-resistant sensor nodes is absolutely necessary. However, such tamper-resistant schemes come at a higher manufacturing cost and are restricted to applications that are not only critical, but use fewer nodes; they should be able to do the following when implementing a security suite (check all tasks completed):

_____**1.** *Authentication:* The main objective of authentication is to prevent impersonation attacks. Hence, authentication can be defined as the process of assuring that the identity of the communicating entity is what it claims to be.

_____**2.** *Integrity:* The goal of integrity is to affirm that the data received is not altered by an interceptor during communication (by insertion, deletion, or replay of data) and is exactly as it was sent by the authorized sender. Usually, cryptographic methods such as digital signatures and hash values are used to provide data integrity.

_____**3.** *Confidentiality:* The goal of confidentiality is to protect the data from unauthorized disclosure. A common approach to achieving confidentiality is to encrypt user data.

_____**4.** *Availability:* The goal of availability is to ensure that the system (network) resources are available and usable by an authorized entity, upon its request. It tries to achieve survivability of the network at all times.

_____**5.** *Access control:* The goal of access control is to enforce access rights to all resources in its system. It tries to prevent unauthorized use of system and network resources. Access control is closely related to authentication attributes. It plays a major role in preventing leakage of information during a node-compromise attack. One of the conventional approaches to access control is to use threshold cryptography. This approach hides data by splitting it into a number of shares. To retrieve the final data, each share should be received through an authenticated process.

_____**6.** *Nonrepudiation:* Nonrepudiation can be best explained with an example. Let Alice and Bob be two nodes, who wish to communicate with each other. Let Alice send a message (M) to Bob. Later, Alice claims that she did not send any message to Bob. Hence, the question that arises is how Bob should be protected if Alice denies any involvement in any form of communication with Bob. Nonrepudiation aims to achieve protection against communicating entities that deny that they ever participated in any sort of communication with the victim.

to the input frequency. In the frequency domain, the output signals appear as noise. Since the pseudorandom code provides a wide bandwidth to the input data, it allows the signal power to drop down below the noise threshold without losing any information. Therefore, this technique is hard for an eavesdropper to detect, due to lower energy levels per frequency and more tolerance to interference. The above-mentioned schemes can provide security only as long as the hopping pattern or the spreading code is not disclosed to any adversary.

Security Measures in the Data Link Layer

Link-layer security plays an important role in providing hop-by-hop security. Its protocols are useful in handling fair channel access, neighbor-node discovery, and frame error control. Legacy security protocols such as Secure Socket Layer (SSL) or Internet Protocol Security (IPSec) cannot be applied directly to WSN because they do not provide data aggregation or allow in-network processing, which are prime requirements in designing security protocols.

To prevent denial-of-service (DoS) attacks on WSN, it is proposed that each intermediate node in the active routing path perform an authentication and integrity check. However, if a few intermediate nodes in the active path have very low energy levels, and if they are forced to perform authentication checks, they will expend all their energy and disrupt the active path. On the other hand, if we look at end-to-end authentication in WSN, it is more energy-efficient, since the sink node (resource-abundant) is the only node that performs authentication and integrity checks. Nevertheless, this scheme is vulnerable to many types of security attacks (black hole, selective forwarding, and eavesdropping). Hence there is a need for adaptive schemes that consider the energy levels of each node when deciding on the authentication schemes.

Early security approaches focused on symmetric keying techniques, and authentication was achieved using Message Authentication Code (MAC). One of the common MAC schemes is a cipher-block chaining message authentication code. However, this scheme is not secure for variable-length input messages. Hence the end user (sensor nodes) has to pad the input messages to be equal to a multiple of the block cipher. Therefore, each node has to waste energy, padding input data. To overcome this issue, other block cipher models such as CTR and OCB have been proposed. With reference to confidentiality, symmetric encryption schemes used to protect WSN are DES, AES, RC5, and Skipjack (block ciphers) and RC4 (a stream cipher). Usually, block ciphers are preferred to stream ciphers because they allow authentication and encryption.

A few proposed link-layer security frameworks include TinySec, Sensec, SNEP, MiniSec, SecureSense [9,10], and ZigBee Alliance [11]. However, these schemes have limitations. For example, in TinySec a single key is manually programmed into all the sensor nodes in the network. A simple node-capture attack on any one of these nodes may result in the leakage of the secret key and the compromising of the entire network. A stronger keying mechanism is needed to secure TinySec. In addition, TinySec requires padding for input messages that are less than 8 bytes. It uses block cipher to encrypt messages, and for messages that are less than 8 bytes, the node will have to use extra energy to pad the message before encrypting.

3. SECURITY MEASURES FOR WSN

In all WSN applications, authentication and further encryption are fundamental security requirements and are useful in mitigating impersonation attacks. They are also useful in preventing the ever-increasing DoS and DDoS attacks on limited resource-constraint environments such as WSN.

Authentication

Three scenarios exist in WSN that require authenticated communication:

- Sink node to sensor nodes and vice versa
- Sensor node with other sensor nodes
- Outside user and sensor nodes

Most of the time critical applications in WSN require a message to be sent as promptly as possible. The intermediate nodes between the sender and receiver are responsible for relaying the message to the receiver. If one of the nodes is compromised, the malicious node can inject falsified packets into the network while routing messages. Such an act could lead to falsified distribution of such messages and, in turn, deplete the energy levels of other honest nodes. Hence, there is a need to filter messages as early as possible by authenticating every message, consequently conserving relaying energy.

In most WSN applications, the sensor nodes are expected to aggregate, process, store, and supply sensed data upon the end user's query. For example, in a military application, soldiers would require constant interaction with motion sensors that detect any movement along the border. In such situations, a large number of mobile or static end users could query the sensor nodes for sensed data. Usually, such interactions are realized through broadcast/multicast operations. Therefore, in such situations, a broadcast authentication mechanism is required before the query is sent. Furthermore, access control is also required, which would only allow the authorized user to access data to which he is entitled. Broadcast

authentication was first addressed in μTESLA [9]. In this scheme, users are assumed to be a few trustworthy sink nodes. This scheme uses one-way hash functions, and the hash pre-images are used as keys to the Message Authentication Code (MAC) algorithm.

However, the messages are transmitted through a wireless medium, which consumes a considerable amount of time. In addition, the hop-by-hop routing nature of WSN further creates a delay in transmission. Hence, there is an increased need for rapid generation and verification of signature schemes.

The existing symmetric schemes such as μTESLA and its variants use Message Authentication Code (MAC) to gain efficiency in terms of processing and energy consumption. However, these symmetric schemes suffer from delayed authentication and sluggish performance for large-scale networks, and they are susceptible to DoS attacks due to late authentication. Furthermore, multiple senders cannot send authenticated broadcast messages simultaneously. For example, if a single node is interested in broadcasting a message, it would have to send a Unicast message to its respective sink node, which would then broadcast the message to all the other nodes on its behalf. Because of resource constraint, asymmetric schemes—for example, digital signatures that would require public key certificates—were pronounced inefficient. Hence, to address this problem, new avenues are being explored to introduce authentication in public-key cryptography in WSN [12].

Lightweight Private Key Infrastructure (PKI) for WSN

Although the applicability of PKI-based approaches has been deemed inappropriate for a resource-constraint environment such as WSN, security researchers have been proposing new lightweight PKI-based approaches for WSN. For instance, a simplified version of Secure Socket Layer (SSL) has been proposed in WSN [13]. Although this SSL version has a smaller overhead when compared to the usual SSL/TLS protocol, it is still not directly applicable to mobile sensor nodes because it would lead to increased communication and computational overhead. For instance, in an ad-hoc mobile sensor network, the nodes keep changing their location, and any change in their position would compel them to initiate the SSL protocol before informing their neighbors of their new location. In addition, schemes such as TinyPK have been designed that are in conjunction with TinySec and facilitate authentication and key agreement between sensor nodes [14]. However, TinyPK implements the Diffie-Hellman key-exchange protocol, which is susceptible to an active man-in-the-middle attack. Huang et al. [15] proposed a hybrid architecture for authenticated key establishment of a session key between a leaf node and a sink node or an end user. This protocol leverages on the difference in the computational and communication capabilities between the leaf node and the resource-abundant device (sink node or end user). During the inception of the protocol, both parties exchange certificates issued by a certificate authority (CA) to extract each other's public keys. However, the corresponding private keys are discovered after both parties run the protocol. This step in this protocol can easily be exploited by an adversary by replaying a valid certificate that would result in a DoS attack. As a result, the nodes are forced to perform expensive computations and waste their resources and bandwidth. In addition, [16] showed a serious vulnerability in Huang et al.'s scheme wherein an end user can easily discover the long-term private key of a leaf node after having one normal run of the protocol.

To expunge the transmission of public key certificates, Ren, Lou, and Zhang [17] propose a Hybrid Authentication Scheme (HAS) for a multiuser broadcast authentication scheme in WSN. In this scheme, each sensor node is preloaded with the required public key information of the end user using the Bloom filter and Merkle hash tree [18,19]. However, HAS with the Merkle hash tree does not facilitate user scalability (a new user can only be added into the network after revocation of the old user).

Key Management in WSN

Recent advances in Integrated Circuit (IC) fabrication have led to the proliferation of wireless sensor networks, which comprise low-cost sensors with limited storage and processing power. WSNs have applicability in diverse fields, such as military, ocean, and wildlife monitoring; earthquake monitoring; safety monitoring in buildings; and in new smart home technology proposed by 4 G technologies. However, such networks deviate from the legacy-embedded wireless networks in terms of scalability, dynamic nature with regard to the addition or deletion of nodes, and deployment areas. Hence, there is a greater challenge in providing security by taking such harsh operational requirements into consideration. One such challenge is in the area of key distribution and its management. In addition, the lack of a-priori information about the topology of WSN makes key management fairly complex. Key distribution provides communication secrecy (confidentiality) and authentication among sensor nodes, and key revocation refers to the task of removing compromised keys from the network. Key distribution can be further divided into symmetric and asymmetric key-distribution protocols.

In recent years, considerable work has been done in proposing new symmetric key-distribution protocols in WSN, but less effort has been invested in the area of asymmetric key-distribution algorithms in WSN, which have low computational and storage requirements. Of late, significant work has been done to show the applicability of implementing binary-field algorithms on sensor nodes [20]. Consequently, such implementations have resulted in considerable reductions in computational time and memory access.

In general, key-distribution schemes in WSN can be broadly classified into four classes: symmetric key algorithms, trusted server mechanisms, random key-predistribution schemes, and public key algorithms. Later in this chapter, we review a few existing key-distribution schemes in WSN.

Symmetric Key Algorithms

In this class, a single shared key is used to perform the encryption and decryption operations in a communication network.

Fully Pairwise-Shared Keys

In this scheme, every node in the network shares a unique, preshared, symmetric key with every other node in the network. The keys are preloaded into the sensor nodes before deployment. Hence, in a network of n nodes, there would be a total of $n(n-1)/2$ unique keys. Subsequently, every node stores $n-1$ keys, one for each of the other nodes in the network. In this class of protocols, the compromise of a few sensor nodes will not result in the complete collapse of the entire network. However, the applicability of this approach in large sensor networks is not pragmatic, as each node would need to store $n-1$ keys, thus resulting in the rapid exhaustion of its limited memory space. In addition, nodes usually communicate with their immediate one-hop neighbors, thereby eliminating the need to establish unique keys with every node in the network. Although symmetric key algorithms are limited in terms of key distribution, they provide basic cryptographic primitives, which can be used in combination with asymmetric key cryptographic algorithms.

Trusted Server Mechanisms

In this category, key distribution is done via centralized trusted servers, which are usually static in nature. In WSN, the sink node or the base station can act as a key-distribution center (KDC). Usually, unique symmetric keys are shared between the sink node and the ordinary nodes. If two nodes were to communicate with each other, they would first authenticate with the base station, after which the base station generates a link key and sends it securely to both parties.

An example of a base-station-mediated key-agreement protocol is the Security Protocol for Sensor Networks: SPINS [9]. Using this protocol, one can preload only one unique single key in every node of the network. Hence, a node capture will not result in the total compromise of the network. In addition, centralized revocation is possible through authenticated unicasts from the trusted base station. The main drawback of this scheme is that the trusted base station represents a single point of compromise for security information, and may also induce a focused communication load centered on the base station, which may lead to early battery exhaustion for the nodes closest to the base station. Another concern is that certain networks do not have a suitable, highly functional, and tamper-proof device that can be used as a secure KDC.

λ-Secure $n \times n$ Key-Establishment Schemes

Now let's address the problem of key distribution and key establishment [21,22] between all pairs of n principals. Although these schemes were originally intended for group keying in traditional networks, and not for sensor networks, they are included here because of their relevance to the development of subsequent key-distribution schemes for sensor networks. The schemes of both Blom and Blundo et al. have an important resiliency property—the λ-secure property: The coalition of no more than λ-compromised sensor nodes reveals nothing about the pairwise key between any two noncompromised nodes.

The main advantage of this class of schemes is that they allow a parameterizable trade-off between security and memory overhead. Whereas the full pairwise scheme involves the storage of $O(n)$ keys at each node and is n-secure, this class of schemes allows the storage of $O(\lambda)$ keys in return for a λ-secure property, and it is perfectly resilient to node compromise until $\lambda + 1$ nodes have been compromised, at which point the entire network's communications are compromised.

Random Key-Predistribution Schemes

In this method, keys are predistributed by preloading random keying material on sensor nodes with the intention of establishing a common secret key between the communicating entities. Upon deployment, these nodes carry out a lookup process to see if a shared key exists between them. As keys are preloaded in a random manner, a certain set of nodes may not share a common key with each other. In such cases, nodes could make use of their immediate neighbors who share keys as bridges between the nodes that do not share a common key. One of the early

key-sharing algorithms using random graph theory was proposed by Eschenauer and Gligor [23].

Basic Random Key-Predistribution Scheme

In this scheme, let m denote the number of distinct cryptographic keys that can be stored on the key ring of a sensor node. This scheme is divided into three phases as follows.

Phase I: Key Predistribution

In this initialization phase, a random pool (set) of keys Q are picked from the total possible key space. In addition, for each node, m keys are randomly selected from the key pool Q and stored into the node's memory. Each of the m keys has identifiers that will be used to map the keys by the receiving nodes during the discovery phase of this scheme (discussed next). This set of m keys is called the node's key ring. The number of keys in the key pool $|Q|$ (key pool size) is chosen such that any two random subsets of size m in Q will share at least one key, with some probability p.

Phase II: Shared-Key Discovery

On deployment, neighboring sensor nodes begin the discovery process to find out if they share a common key with each other; if they do, then they establish a secure link. There could be many modes for the discovery phase, such as broadcasting the list of identifiers existing in their key ring in clear text or through a challenge-response mechanism. If the probability p were chosen correctly for the network's neighbor density, then the resultant graph of secure links would be connected with some high probability. The remaining links in the graph are then filled in by routing key-establishment messages, along this connected network of initial secure links. From a security perspective, although this approach does not reveal any important information to the adversary, it is still susceptible to a passive traffic analysis attack.

Phase III: Path-Key Establishment

Upon completing the discovery phase, if two nodes in the network discover that they do not share a key between them, they send an encrypted message to neighbors with whom they share a key, with a request to secure connection with the unshared node. This model assumes that after the completion of *Phase II*, there exist many keys in each key ring that can be used for third-party path-key establishment. Hence, the neighboring nodes generate pairwise keys for nodes that do not directly share a key.

Let us now find this probability p that any two nodes with key ring sizes m in the network share at least one common key from the pool Q. Let p' be the probability that two nodes do not share a key between them. Then, p is defined as

$$p = 1 - p' \tag{16.1}$$

In this case, keys from the key ring are drawn from Q without replacement. The total number of possible key rings t_1 is as follows:

$$t_1 = \frac{Q!}{m!(Q-m)!} \tag{16.2}$$

Now, the total number of possible key rings that do not share a key with a particular key ring t_2 is the number of key rings drawn from the remaining Q-m unused keys in the pool:

$$t_2 = \frac{(Q-m)!}{m!(Q-2m)!} \tag{16.3}$$

Then, the probability that no key is shared between any two rings is t_2/t_1. Hence, the probability p is

$$p = 1 - \frac{t_2}{t_1} = 1 - \frac{((Q-m)!)^2}{Q!(Q-2m)!} \tag{16.4}$$

Usually, the value of p is very large in comparison to m, and using the Sterling's approximation for $n!$, the value of p is

$$p = 1 - \frac{\left(1-\frac{m}{Q}\right)^{2(Q-m+0.5)}}{\left(1-\frac{2m}{Q}\right)^{(Q-2m+0.5)}} \tag{16.5}$$

Figure 16.4 shows the value of p for different values of Q and m. We observe that with the increase in Q, there is a negligible increase in the key ring size m for the same value of p. For example, for $p = 0.5$ and $Q = 6000$, the value of $m = 68$. Subsequently, if the pool size is increased to 10,000, for the same value of $p = 0.5$, m is only increased to 95.

In this scheme, all nodes use the same key pool Q. This implies that the security of the network is gradually eroded as keys from Q are compromised by an adversary that captures more and more nodes. In this scheme, the number of exposed keys is roughly linear to the number of nodes compromised. This characteristic of the basic scheme motivated development of key-predistribution schemes that have better resiliency to node capture. The basic scheme was extended by the q-composite scheme proposed by [24].

q-*Composite Scheme*

In a q-composite key scheme, instead of designing for a given probability p of sharing a single key, the parameters are altered such that any two nodes have a given probability p of sharing at least q different keys from the key

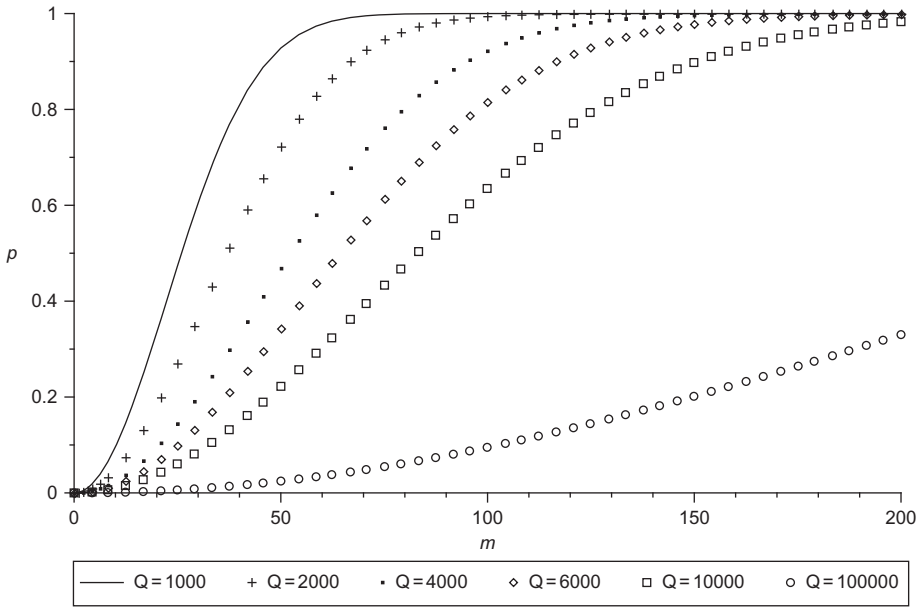

FIGURE 16.4 Probability of sharing at least one shared key using Eschenauer and Gligor's scheme.

pool. All q keys are used in the generation of the key, which encrypts communications between sensor nodes; hence, to eavesdrop on the secured link, the adversary now has to compromise all q keys, instead of just one. As q increases, it is exponentially harder for the attacker to break a link by taking possession of a given set. However, increasing the probability of overlap in this fashion naturally involves reducing the size of the key pool Q. Thus, the smaller key-pool size makes the scheme more vulnerable to an adversary that is capable of compromising larger numbers of sensor nodes.

The key-predistribution phase of this model is similar to *Phase I*, which is discussed later in the chapter, with the only exception being the key-pool size Q. In the shared key-discovery phase, each node must find nodes that share all common keys with each other. The discovery mechanism is similar to that of *Phase II*. Although a broadcast-based approach is susceptible to an eavesdropping attack, alternative methods that are slower but more secure are suggested where the nodes use the Merkle puzzle for key discovery [18]. After the discovery phase, each node would be able to recognize its immediate neighboring nodes with which it would share at least q keys. Subsequently, each node could establish a link between nodes that share at least q keys by hashing the keys in some canonical order. For example, $K = \text{hash}(k_1 \,||\, k_2 \,||\, k_3 \,||\, \ldots \,||\, k_q)$.

In this scheme, the key pool size $|Q|$ plays a critical role because with a larger Q, the probability of any two nodes sharing at least q keys would be much less. Consequently, after bootstrapping, the network may not be connected. On the contrary, if $|Q|$ is small, the security of the network is compromised. Hence, $|Q|$ should be such that the probability of sharing at least q key should be greater than or equal to the probability of successfully achieving a key setup with any of its neighbors. The approach used to calculate the probability of any two nodes sharing exactly i keys $p'(i)$ is similar to calculating p, as shown in Eq. (16.4), and is given as

$$p'(i) = \frac{\binom{|Q|}{i}\binom{|Q|-i}{2(m-i)}\binom{2(m-i)}{m-i}}{\binom{|Q|}{m}^2} \tag{16.6}$$

For example, in Figure 16.5, we find the value of $|Q|$ for a given m and i. In this case, for $m = 200$ and $i = 10$, we achieve a maximum $p'(i)$for $|Q| = 3900$.

In general, random key predistribution presents a desirable trade-off between the insecurity of using a single network-wide key and the impractical high memory overhead of using unique pairwise keys. Its main advantage is that it provides much lower memory overhead than the full pairwise key scheme, while being more resilient to node compromise than the single-network-wide key scheme. Furthermore, it is fully distributed and does not require a trusted base station. The main disadvantages of this approach are the probabilistic nature of the scheme, which makes it difficult to provide the guarantee of the initial graph of secure links being connected under nonuniform conditions or sparse deployments. Furthermore, since keys can be shared between a large number of nodes, this class of schemes does not provide very high resilience against node compromise and subsequent exposure of node keys.

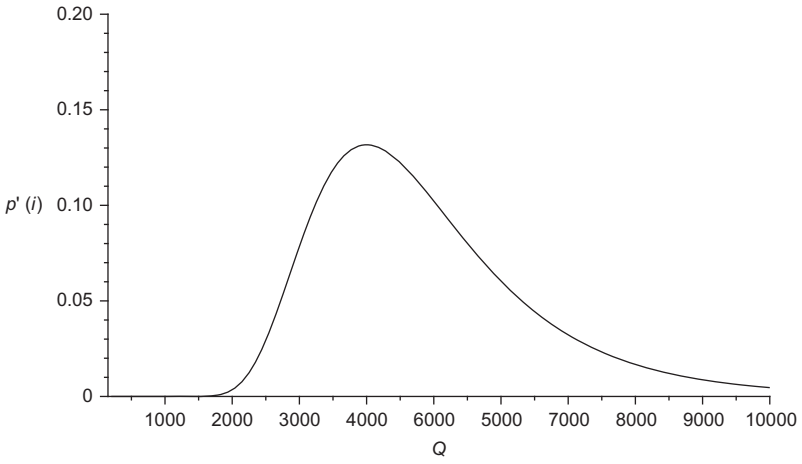

FIGURE 16.5 Key-pool set $|Q|$ selection based on $p'(i)$ for $m = 200$ and $i = 10$.

Random Pairwise Key Scheme

The random pairwise key scheme proposed by [24], is a hybrid of the random key-predistribution scheme and the full pairwise key scheme. In the analysis of random key predistribution, it was deduced that as long as any two nodes can form a secure link with at least a probability p, the entire network will be connected with secure links with high probability. Based on this observation, Chan noted that it is not necessary to perform full pairwise key distribution to achieve a network where any two nodes can find a secure pathway to each other. Instead of preloading $n-1$ unique pairwise keys in each node, the random pairwise key scheme preloads $m \ll n$ unique pairwise keys from each node. The m keys of a key ring are a small, random subset of the $n-1$ possible unique keys that this node could share with the other n nodes in the network. By using the same reasoning as the random key-predistribution scheme, as long as these m keys provide sufficient probability p of enabling any two neighboring nodes to establish a secure link, the resultant graph of initial secure links will have a high probability of being connected. The remaining links are then established using this initial graph exactly as in the random key-predistribution scheme.

Chan et al. (2003) present a preliminary initial distributed-node-revocation scheme that makes use of the fact that possessing unique pairwise keys allows nodes to perform node-to-node identity authentication. In their scheme, each of the m nodes that shares a unique pairwise key with the target node (the node's participants) carries a preloaded vote that it can use to signify a message that the target is compromised. These m votes form a Merkle hash tree with m leaves [18]. To vote against the target node, a node performs a network-wide broadcast of its vote (its leaf in the Merkle hash tree) along with the log m internal hash values, which will allow the other participants of the target to verify that this leaf value is part of the Merkle hash tree. Once the t participants of a given target have voted, and the votes have been verified by the other m participants using the Merkle hash tree, all m nodes will erase any pairwise keys shared with the target, thus revoking it from the network.

The random pairwise key scheme inherits both strengths and weaknesses from the full pairwise key scheme and the random key-distribution scheme. Under the random pairwise key scheme, the nodes captured do not reveal information to the rest of the network, and central revocation can be accomplished by just unicasting to each of the nodes that share keys with the revoked node. It also involves a much lower memory overhead than the full pairwise keys scheme. Unfortunately, like the random key-predistribution schemes, it is probabilistic and cannot be guaranteed to work in nonuniform or sparse deployments.

Multispace Key Schemes

This class of schemes is a hybrid between random key predistribution and the λ-secure $n \times n$ key-establishment schemes. (These schemes were first proposed by [25].) Recall that in random key predistribution, a key pool is first selected from the universe of possible keys. Each sensor node is then preloaded with a set of keys from the key pool such that any two nodes possess some chosen probability p of sharing enough keys to form a secure link. Multispace key schemes use the same basic notion of random key predistribution but use key spaces, where individual keys are used in random key predistribution. Hence, the key pool is replaced by a pool of key spaces, and each node randomly selects a subset of key spaces from the pool of key spaces, such that any two nodes will have some common key space with probability p. Each key space represents a unique instance of a different λ-secure $n \times n$ key-establishment scheme [21]. If two nodes possess the same key space, they can then perform

the relevant λ-secure $n \times n$ key-establishment scheme to generate a secure session key.

The main advantage of multispace schemes is that node compromise under these schemes reveals much less information to the adversary than occurs with the random key-predistribution schemes. However, they retain the disadvantage of being probabilistic in nature (no guarantee of success in nonuniform or sparse deployments). Furthermore, they experience the threshold-based sudden security failure mode that is a characteristic of the λ-secure schemes. Other schemes have combined λ-secure schemes with constructions other than random key-space selection. Liu and Ning [26], in particular, describe a deterministic grid-based construction in which key spaces are used to perform intermediary-based key establishment between nodes.

Deterministic Key-Predistribution Schemes

One drawback of the random key-distribution approach is that it does not guarantee success. [27], as well as [28], propose the use of combinatorial design techniques to allocate keys to nodes in such a way as to always ensure key sharing between any two nodes. The amount of memory required per node is typically some fractional power of the overall supported network size ($O(\sqrt{n})$). The main drawback of these schemes is that the same keys are shared between many nodes, leading to weaker resistance to node compromise. [24] have proposed a deterministic scheme using peer nodes as intermediaries in key establishments with similar memory overheads; compared with the combinatorial design approach, this scheme trades off increased communication cost for greater resistance against node compromise.

Public Key Algorithms

Although these algorithms are based on asymmetric key cryptography and are more resource intensive than symmetric key algorithms, they offer better security services, which are much needed and highly advantageous in WSN. As a result, there is motivation to pursue research in developing secure and efficient key-distribution mechanisms suitable in a resource-constraint environment such as WSN. Most of the implementations use Rivest, Shamir, Adleman (RSA) or elliptic curve cryptography (ECC) [12,20].

For example, TinyPK uses the Diffie-Hellman key-exchange technique for key agreement between nodes and is based on the legacy RSA cryptosystem. The main motive of this protocol is to facilitate secure communication between external users and the sensor networks. The external user's identity is established by a CA, where his or her public key is signed by the CA's private key.

Considering the state of the art in large-number factorization, key-size values are usually set to 1024 bits in RSA as lower values are considerably vulnerable to security attacks. In addition, the public key exponent e is set to 3, and all the resource-intensive operations are carried out on external servers. In this model, resource-abundant devices bear the burden of RSA private key operations, and, hence, the sensor nodes maintain higher energy levels during operations.

4. SECURE ROUTING IN WSN

Routing is one of the most fundamental operations in any network that attempts to ensure the delivery of messages from a source to a selected destination. It is a two-step method that involves the process of discovering a suitable route between the concerned source and its destination, and the forwarding of messages using this discovered route. In traditional networks (IP or 3 G networks), routing operations are dedicated to special nodes, such as routers. However, WSNs consist of resource-constraint devices operating in an ad-hoc decentralized manner that requires all the network operations to be done by these ordinary sensor nodes. Some real-time applications (remote-sensing operations) require the routing protocols to facilitate the timely delivery of messages. However, such applications are too resource intensive in WSN and require routing protocols that can balance the energy consumption of the entire network. Furthermore, the number of nodes operating in a WSN scenario is much larger than conventional networks. Consequently, there is a need for the mass production of low-cost nodes. However, with the increase in the number of sensor nodes to meet the current demand for sensor applications, construction of each node to be tamper resistant would be very expensive. As a result, nodes could be susceptible to a node-capture attack. Hence, routing protocols used in traditional networks cannot be applied directly to a resource-constraint environment such as WSN. As a result, new arrays of routing protocols have been designed for WSN [29].

5. ROUTING CLASSIFICATIONS IN WSN

Routing protocols in WSN can be classified by several criteria. Such criteria would be data centricity, location information, network layering and in-network processing, path redundancy, a Quality of Service (QoS) requirement, and network heterogeneity.

Datacentric Communication

Conventional networks such as IP networks use a node-centric routing model in which information is exchanged

using a unique addressing scheme (IP version 4 or 6 or higher). Based on the route the query took to reach the destined node, each source node independently sends data via the shortest path to the concerned sink node. In contrast, a datacentric model is more focused on the aggregated data rather than on identifying the exact node's identifiers. Although the request/response scheme is similar in both of the models, the sink node or cluster head initiates a request for interested data and the responsible nodes respond with the requested data; they vary in the manner in which the nodes send data back to the sink node or cluster heads. The intermediate routing nodes inspect the data that is being sent to the sink node and perform some form of consolidation operation, such that the sink node receives aggregated data from different sources. Figures 16.6 and 16.7 illustrate the distinction between address-centric and datacentric models in WSN. Figure 16.6 shows the address-centric model in which two sources (nodes C and E) send information to the sink node via the shortest path. Node C sends via node A, and node E sends via nodes D and B.

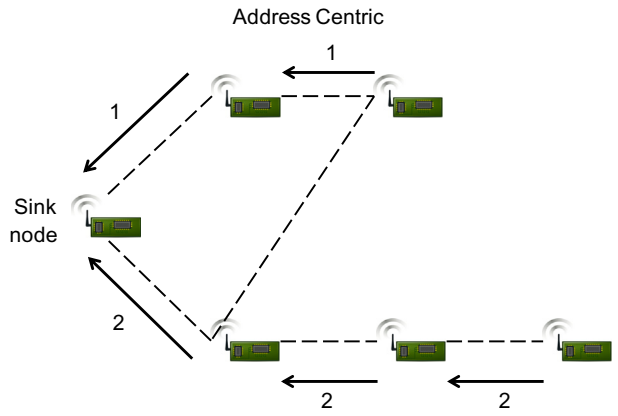

FIGURE 16.6 Address-Centric communication in WSN.

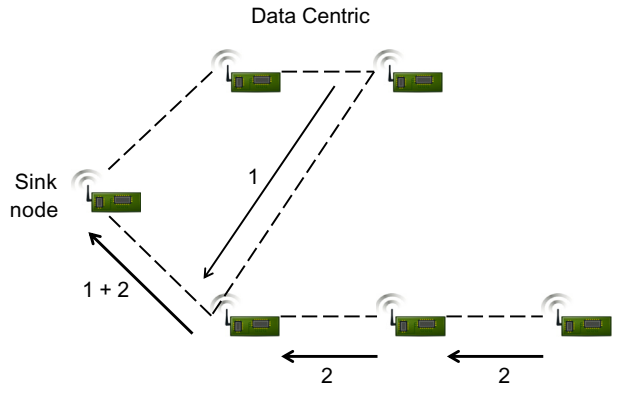

FIGURE 16.7 Datacentric communication in WSN.

In contrast, Figure 16.7 shows the datacentric model in which data from node C is directly sent to node B for consolidation, and, subsequently, the aggregated data is sent to the sink node. In cases where node C cannot directly contact node B, an intermediate node closer to node B is responsible for forwarding the data to node B. On comparing both models, the datacentric model is more energy efficient because only four messages are utilized in sending information from two different sources to the sink node, as opposed to five messages in the address-centric model (Figure 16.6).

Location Information

The physical location of a node in a network is an essential metric for designing routing protocols in a WSN. For example, the protocols could be designed for short- or long-range communication, depending on the position of the nodes. In addition, the positions of the nodes influence the design of the forwarding mechanism, which, in turn, affects the overall energy consumption of the system. In situations in which routing tables or global knowledge of the network is not required, location-based routing could be useful from a scalability point of view. However, one of the major challenges with such networks is that each node should be aware of its position with reference to the sink node. Additionally, the use of Global Positioning System (GPS)-based chips could weigh heavily on the limited resources available in a sensor-network environment, leading to an increase in the price and energy consumption of the system.

Network Layering and In-Network Processing

The architecture of a network could be flat, in the sense that all sensors have the same role. In other words, all sensors forward their sensed data to the sink without necessarily passing through a particular node. A network is said to be *nonlayered* if all sensors form only one group in which the sensors collaborate to accomplish a common monitoring task. On the other hand, the sensors in a network can be grouped into *clusters*, each of which is managed by a specific sensor called a *cluster head*. These types of networks are considered to be *layered*, and, any sensed data should pass through one or more cluster heads before reaching the sink. These cluster heads are supposed to be powerful enough to process the data they receive before sending it to the sink.

All other sensors only need to sense the environment and send their data to the cluster heads for further processing. In some sensing applications, redundancy and correlation exist in the gathered data. Hence, it would be desirable to only transmit more representative data. For

example, in monitoring the temperature of a room, the variation in the data within a given region is expected to be small. Thus, the sink is not interested in receiving all the temperature measures, but rather only some of them. This can significantly reduce the communication overhead introduced by data forwarding and improve network performance. In addition, the concept of layering makes a network more scalable and leads to more efficient usage of the energy of sensors, thus extending the network's lifetime.

Extending the network's lifetime is an ultimate goal in the design of a WSN. Given that most of the energy of a sensor is mainly consumed in processing, sensing, and communication, an efficient design approach should take into account these three components of energy consumption. A question that network designers are mostly concerned about is, *How can the lifetime of a network be extended?* To address this problem, several energy-efficient routing and data-dissemination protocols have been proposed that focus on how to forward the data until they reach the sink, regardless of the type of data being transmitted from the source sensors to the sink. Among those protocols, one class does not update the data at the intermediate sensors. That is, each intermediate sensor only acts as a pure data relay without altering any of the data it has received. Another class of protocols introduces the concept of *in-network processing* to handle unnecessary redundancy and correlation contained in the sensed data.

In many applications, the data sensed by the sensors has a certain amount of redundancy and correlation. It would be desirable if the sink could only receive relevant data, for faster and better decision making. For this purpose, the sensed data should be processed at intermediate sensors before reaching the sink. The benefit of this in-network processing, such as data fusion, can be seen when vector data rather than scalar data are being transmitted. For example, in an application monitoring the temperature of a room, the sensed data is scalar (integer or real values). Hence, the cost of data communication is not very high, and the data fusion or aggregation is not as costly. But continuously sending unnecessary and redundant data will consume a huge amount of energy. If a sensing application has to send a large amount of data, for example, images, to the sink for further analysis and processing, it would consume a huge amount of energy. In this case, it would be more beneficial if those images, sensed by different sensors, could be aggregated and only a few of them sent. However, it is also true that processing those images for data fusion requires a considerable amount of energy. Moreover, there will be a delay due to the processing of those images. Therefore, there is a trade-off between data communication and fusion, in these types of information-intensive networks, where the sensed data is not scalar but rather vector.

Path Redundancy

The design of WSN not only should consider scalability and energy efficiency but should also be robust in nature, which means that a network remains operational despite the occurrence of sensor-node and link failures. The reasons for the failure could be intentional (security attack) or unintentional (defective node or natural calamity). One of the approaches to make the system more robust is to incorporate multipath routing. In short, multipath routing implies the existence of multiple paths (disjoint or partially disjoint) between source node and destination sensors (cluster heads or sink node) [30].

Although maintaining alternate paths in a routing table introduces some overhead and consumes more energy, multipath routing is an effective technique to improve robustness when link failures occur. Link failures could occur for many different reasons, such as frequent topological changes due to unreliable wireless communication links. Moreover, multipath routing enables recovery from sensor and link failures and provides the necessary resilience to the network at the cost of excessive redundancy.

Quality of Service (QoS)

WSN applications have varied requirements that are usually expressed in terms of some metrics, such as delay, fault tolerance, and reliability. For example, real-time applications (video surveillance) are dependent on delay bounds.

Hence, for such applications, the sensed data must reach the sink within a certain time. In addition, a desired property of WSN applications is fault tolerance, which means that a network should remain functional in the event of sensor or link failures. Another desired property is reliability, wherein the aggregated data should be received by the sink, as correctly as possible. This would ensure accurate decision making by the sink node. However, metrics such as fault tolerance and reliability necessitate the deployment of additional sensors, yielding additional energy consumption, so that the network can recover swiftly and deliver accurate sensed data to the sink, despite some sensor or link failures. Hence, routing and data-dissemination protocols should consider the trade-offs between fault tolerance, reliability, energy, and delay. Recall that energy is a constraint that should be considered by any routing and data-dissemination protocol to guarantee efficient usage of the amount of energy available at each sensor.

Network Dynamics

Requirements such as limited-energy use (discussed previously) and goals such as mobility have had direct impact on the design decisions of WSN network topology. In theory, a deterministic sensor deployment approach would provide even coverage of the area that has to be sensed. In addition, this approach would require fewer sensor nodes for accomplishing the required sensing task. However, in a real environment with an uneven terrain, it can be extremely challenging to apply a deterministic sensor deployment strategy. As a result, we are only left with the option of distributing the nodes in a random fashion. Consequently, not all areas of the sensing region are evenly covered by the sensors, thus leading to a coverage hole. In addition, there is a possibility of not all sensor nodes in the network being connected with each other or even with the sink node. In such situations, mobility plays an important role and becomes the main source of network dynamics that can be used to solve problems. In any sensor network, the aggregated data will be transmitted over some established paths between the source sensors and the cluster heads or sink node. And the establishment of optimal paths depends on whether the sensors are static or mobile. Hence, routing and data-dissemination protocols can be classified based on whether a network is static or dynamic.

In a static network, every node in the network is static—that is, both the sensors and the sink node remain in their fixed positions during their collaborative operation of monitoring a physical environment. Therefore, there is not much overhead required to maintain routes between the sensors and the sink and between the sensors themselves. In particular, the positions of the sensors and the sink can be learned before data exchange by exchanging some control messages. In certain cases, if the terrain is familiar, node positions can be preconfigured in nodes before deployment. Furthermore, neighbors of a given sensor do not change unless a new sensor has joined the network or an existing sensor has left the network, either by its own will or because of exhaustion of its battery life.

In a mobile network, either the sensors or the sinks or cluster heads are moving. As a result, the routes between the sensors and the sink are changing frequently in such a dynamic environment. Hence, a currently active route could at any time become inactive. This route instability would result in additional overhead and delay in discovering valid routes for data transmission and forwarding. To overcome this drawback, routing algorithms have been proposed in which the ordinary sensors and sinks are designed to be static, whereas certain relay nodes such as cluster heads could be mobile. One such example is the mobile ubiquitous LAN extensions (MULES)-based architecture [31].

In conclusion, the need for mobility in WSN is application dependent. For example, in applications that measure temperature, humidity, sound, or light in an enclosed area, there is no need to have mobile sensors or a mobile sink. However, in monitoring a moving object in a battlefield environment, or in monitoring endangered species, there is a need for mobile sensors in the network to efficiently track the object. In such scenarios, it has been observed that the use of mobile relays helps increase the lifetime of a WSN.

Network Heterogeneity

Early research on sensor networks focused on homogeneous network architecture. However, recently heterogeneous sensor networks have experienced increasing popularity because they significantly increase the lifetime and reliability of the system. A heterogeneous sensor network usually consists of a large number of low-cost nodes for the sensing operation and a few resource-abundant nodes that primarily perform data filtering, aggregation, and transport operations. Although heterogeneous networks have gained precedence over homogeneous networks, the efficient realization of heterogeneity in a sensor network requires prior systematic planning for placing these heterogeneous resources in a resource-aware manner [32].

Routing Protocols in WSN

Routing in ad-hoc networks has been very challenging owing to node mobility. Hence, a routing path established in the beginning between the source and the destination may not exist at a later time interval. Furthermore, in a resource-constraint environment such as WSN, the energy levels of the intermediate nodes must be considered in making routing decisions.

Routing protocols in WSN can be broadly classified into proactive, reactive, hybrid, and location-aware routing protocols [33]. In a proactive routing scheme, each node maintains an up-to-date routing table by frequently querying its immediate neighbors for routing information. An example of such a scheme is the Destination Sequenced Distance Vector (DSDV) routing protocol [34]. However, one of the major drawbacks of such schemes is the additional overhead due to frequent routing updates. In contrast, reactive routing involves on-the-fly route establishment and is demand driven. It is based on a request−response model. The initial discovery phase, to find the destined node, could involve flooding, and the response phase establishes the transient active routing path. Examples include Ad-hoc On-Demand Distance Vector (AODV) routing and Dynamic Source Routing (DSR) [35,36].

Various hybrid protocols use the node-discovery method of the proactive routing protocol, along with the on-the-fly routing-path establishment method to produce a hybrid version of the protocol. The Zone Routing Protocol (ZRP) is an example of such a hybrid scheme [37]. In position-aware routing protocols, the nodes select the geographically closest neighboring node when making routing decisions. An example of such a protocol is the Geographic- and Energy-Aware Routing (GEAR) protocol [38]. However, GEAR does not take security into consideration. Most of the security schemes in WSN have focused on symmetric-key cryptography, due to the notion that asymmetric-key cryptography (RSA-based algorithms) was computationally intensive. However, symmetric-key cryptography has major drawbacks with regard to key management, and the security is based on preshared secret keys. With the successful implementation of pairing-based cryptographic algorithms in WSN, a new platform is provided to implement asymmetric-key cryptographic schemes in WSN [20].

Selective-Forwarding Attack in WSN

Many routing protocols in WSN use a breadth-first spanning-tree algorithm to broadcast routing updates [5,39]. The sink node periodically broadcasts updated routing information to its immediate cluster heads. Then, these cluster heads re-broadcast this information to their immediate neighbors, and the process continues recursively. During this process, each intermediate node makes a note of its parent node, where the parent node is the first node that was able to make contact with its subordinate node and relay the routing information. When all the active nodes are operational, they should send all the sensed data to their parent node. However, this protocol is vulnerable to many attacks.

Cross-Layer Design Approach in WSN

Recently, a flurry of cross-layer design schemes have been proposed in WSN. As the fusion of secure networking and wireless communication occupies center stage in sensor networks, the traditional layered protocol architecture on which most of the networks form their basis has come under scrutiny. Although the layered approach has been repeatedly used in wired networks, it has been argued that the same approach cannot be directly applied in resource-constraint, wireless ad-hoc networks such as WSN. To combat this approach, security researchers have proposed several cross-layer design schemes in an ad-hoc environment [40]. Unlike the layering approach, where protocols at each layer are designed independently, cross-layer designs aim at exploiting the dependence between different protocol layers to achieve maximum performance gains. In the current state of the art in the paradigm of cross-layer design schemes in ad-hoc wireless networks, several diverse interpretations exist. One of the main reasons for such varied explanations is that the design effort is largely dominated by researchers who have made independent efforts in designing different layers of the stack. Many of the cross-layer designs depend on other cross-layer designs and hence raise the fundamental question of the coexistence of different cross-layer design proposals. In addition, the question of time synchronization between various cross-layer schemes and the roles each layer of the stack plays is an active area of research. The wireless medium allows richer modalities of communication than wired networks. For example, nodes can make use of the inherent broadcast nature of the wireless medium and cooperate with each other.

Employing modalities such as node cooperation in protocol design also calls for cross-layer design. The goal of designing security solutions with a cross-layer design approach takes us to a new paradigm of security research. The main objective of security solutions in a network is to provide security services such as authentication, integrity, confidentiality, and availability to the users. In wireless ad-hoc networks, due to the unreliable nature of the shared radio medium, attackers can launch varying attacks, ranging from passive reconnaissance attacks to active man-in-the-middle attacks. Routing in WSN is hop by hop and assumes a trusted, cooperative environment as intermediate nodes act as relays. However, compromised intermediate nodes can launch varying routing attacks, such as black-hole, wormhole, flood rushing, and selective-forwarding attacks. In this part of the chapter, we review the existing state of the art in the cross-layer design from a security perspective. In addition, as an example, we look at a cross-layer key-distribution mechanism.

In recent times, several cross-layer design schemes have been proposed. Cross-layer feedback optimization could be implemented on the sink or the sensor nodes. The cross-layer interactions among the layers can be categorized in different ways. For example, lower to upper (violation in the flow control from bottom to top), upper to lower (violation in the flow control from top to bottom), and lower and upper. In all these cases, new interfaces will be created between layers. In addition, cross-layer designs can be categorized by the integration of adjacent layers, design coupling without interfaces, and horizontal calibrations.

Lower to Upper

The requirement of information from the lower layer to the upper layer at runtime results in the creation of a new

interface between these two layers. In this case, the lower layers update necessary information to the appropriate upper layers via the interface. For example, the data link layer is made aware of the transmit power, and the bit error rate information by the physical layer so that it can adjust its error-correction mechanism. Subsequently, the transport layer can inform the application layer about the TCP packet loss, as it would help the upper layer in the stack (application layer) to adjust its transmitting rate. In addition, it should be noted that self-adaptation loops should not be part of a cross-layer design approach, as they do not require new interfaces to be created between the necessary lower and upper layers. For example, in an auto-rate fallback mechanism for rate selection in a wireless networking environment with multirate physical layers, the Medium Authentication Code (MAC) layer rate selection is dependent on the received acknowledgment that is observable at the MAC layer. Hence, this mechanism would not qualify as a cross-layer design approach as there is no need to create new interfaces for rate adoption.

Upper to Lower

The upper layers provide updated information to the necessary lower layers via an interface. For example, if the application layer senses a delay or loss of data, a direct notification to the data link layer by the application layer would help adapt its error correction mechanism. In addition, delay sensitive packets could be treated with priority. As proposed by Larzon, Bodin, and Schelen [41], lower-to-upper information flow is treated as notifications (the lower layer notifies the upper layer about the underlying network condition), whereas the upper-to-lower information flow is treated as hints (upper layers provide hints to the lower layers on the means to process application data).

Lower and Upper

In this case, both the upper and lower layers are at liberty to transmit notifications about their current state and send queries to the other layers. During runtime, layers executing different tasks can collaborate with each other on an iterative loop basis, resulting in a back and forth communication between them. For example, a back and forth information flow between layers is seen in a proposal to solve the multiple access problem for contention-based wireless ad-hoc networks using joint scheduling and suggesting a distributed power-control algorithm for such networks [42]. In addition, direct communication between layers at runtime could indicate the advantage of making the variables at each layer visible to the other layers of the stack. However, one disadvantage of this approach would be in managing the shared memory spaces between the layers when variables and internal states are shared between different layers.

Integration of Adjacent Layers

The formation of a super-layer by combining two or more adjacent layers would result in a new cross-layer design scheme. The resulting layer would simply provide the union of the services that were provided by the individual layers. For example, a collaborative design of the data-link and physical layer would suffice to produce a super-layer. From a network security perspective, a super-layer that combines network and data link layer would help prevent advanced Address Resolution Protocol (ARP) poisoning attacks.

Design Coupling without Interfaces

Coupling two or more layers during the design phase would avoid creating extra interfaces at runtime that could result in a new cross-layer design approach. However, in deployed networks, one of the architectural challenges would be to integrate the coupled layer with already-existing fixed layers.

Vertical and Horizontal Calibration across Layers

Vertical calibration: Vertical calibration refers to the efficient utilization of parameters across different layers of the vertical stack. The parameters set at the application level could dictate terms to the lower layers and vice versa. For example, the transport protocol (TCP or UDP) chosen at the transport layer would assert reliable or unreliable communication and would directly affect the layers below it. Consequently, the joint adjustment at different layers of the vertical stack would result in a more holistic performance of the system than the adjustment of individual parameters.

Horizontal Calibration

Horizontal calibration could be very useful in a resource-constraint environment such WSN. In this case, not only individual parameters pertaining to that layer are taken into consideration, but parameters pertaining to other compatriot layers are also considered. For example, while routing packets, if the network level state of intermediate nodes is taken into consideration, it would be easy to detect nonactive nodes and could subsequently result in an energy-efficient routing protocol. However, challenges do exist in case the participating nodes do not adhere to the same cross-layer approach as the initiating node.

6. WSN SECURITY FRAMEWORK AND STANDARDS

The standardization of wireless sensor networks proceeds along two main directives: the IEEE 802.15.4 standard [43] and ZigBee [11]. The IEEE 802.15.4 standard defines the physical and Medium access control (MAC) layers, and ZigBee defines the network and application layers. In WSN implementations, the two protocol stacks can be combined to provide low data rate and long-lasting applications on battery-powered wireless devices.

IEEE 802.15.4

The IEEE 802.15.4 MAC layer provides marginal support for security, and the advanced security features (key management and authentication) are the responsibility of the upper layers in the WSN protocol stack. In addition, the MAC layer security services assume that the keys are generated, transmitted, and stored by the upper layers in a secure manner.

The security services provided by the MAC layer include access control, data encryption, frame integrity, and sequential freshness. It should be noted that the security features of the MAC layer are optional and that the use of this feature is at the discretion of the applications existing on the application layer.

ZigBee

The ZigBee Alliance is an association of companies working together to develop standards (and products) for reliable, cost-effective, low-power wireless networking [2]. ZigBee is an emerging technology and is being used in a wide range of products and applications across consumer, commercial, industrial, and government markets worldwide. It builds upon the IEEE 802.15.4 standard described previously. The ZigBee specifications provide authentication, data freshness, message integrity, and encryption:

- *Authentication:* Network-level authentication is achieved by using a common network key. This prevents outsider attacks while adding very little in memory cost. Device level authentication is achieved by using unique link keys between pairs of devices. This prevents insider and outsider attacks but has higher memory cost.
- *Freshness:* ZigBee devices use incoming and outgoing freshness counters to maintain data freshness. These counters are reset every time a new key is created. Devices that communicate once per second will not overflow their freshness counters for 136 years.
- *Message Integrity*: ZigBee specifications provide the options of providing 0-, 32-, 64- or 128-bit data integrity for the transmitted messages. The default is 64-bit integrity.
- *Encryption:* ZigBee uses 128-bit Advanced Encryption Standard (AES) encryption. Encryption protection is possible at the network or device level. Network-level encryption is achieved by using a common network key. Device-level encryption is achieved by using unique link keys between pairs of devices. Encryption can be turned off without impacting freshness, integrity, or authentication, as some applications may not need any encryption.

The closest competitor to ZigBee in personal area network technology is Bluetooth. Although Bluetooth claims a much faster data rate (1 Mbps vs. 250 kbps), ZigBee specifies a longer transmission range and is specifically designed for low-power consumption. If Bluetooth is used in modular robotics applications, it requires a central coordinator and is limited to small networks. However, ZigBee does not have this limitation.

7. SUMMARY

Organizations and individuals benefit when wireless sensor networks and devices are protected. After assessing the risks associated with wireless sensor network technologies, organizations can reduce the risks by applying countermeasures to address specific threats and vulnerabilities. These countermeasures include management, operational, and technical controls. While these countermeasures will not prevent all penetrations and adverse events, they can be effective in reducing many of the common risks associated with wireless sensor networks technology.

Finally, let's move on to the real interactive part of this chapter: review questions/exercises, hands-on projects, case projects, and optional team case project. The answers and/or solutions by chapter can be found in the Online Instructor's Solutions Manual.

CHAPTER REVIEW QUESTIONS/EXERCISES

True/False

1. True or False? Although WSNs have gained a little popularity, there are some serious limitations when implementing security.
2. True or False? WSNs operate in a resource-constrained environment and therefore deviate from the traditional Open System Interconnection (OSI) model.
3. True or False? In WSN, threats to privacy can be further classified into reconnaissance.

4. True or False? The man-in-the-middle attack is not one of the classical attacks that can be executed in a WSN environment.
5. True or False? Due to threats to the WSN, some portion of the network or some of the functionalities or services provided by the network could be damaged and available to the participants of the network.

Multiple Choice

1. The middle layer provides one of the following for applications existing in the upper layers:
   A. RC 4 stream cipher
   B. Temporal Key Integrity Protocol (TKIP)
   C. Application programming interface
   D. Message Integrity Code (MIC)
   E. Extensible Authentication Protocol (EAP) framework
2. Which of the following is responsible for flow and congestion control?
   A. Middle layer
   B. Network layer
   C. Transport layer
   D. Data link layer
   E. All of the above
3. What allows organizations to reason about attacks at a level higher than a simple list of vulnerabilities?
   A. Secure on-demand routing protocol
   B. Taxonomy
   C. Message Authentication Code (MAC)
   D. Authenticated Routing for Ad hoc Networks (ARAN)
   E. Destination-Sequenced Distance Vector (DSDV) routing
4. In what type of attack is the attacker able to intercept and monitor data between communicating nodes, but does not tamper or modify packets for fear of raising suspicion of malicious activity among the nodes?
   A. Active attack
   B. Privacy attack
   C. Eavesdropping attack
   D. Man-in-the-middle attack
   E. Passive attack
5. What occurs when an attacker floods the victim with bogus or spoofed packets with the intent of lowering the response rate of the victim?
   A. HELLO flood attack
   B. Denial-of-service attack
   C. Sinkhole attack
   D. Sybil attack
   E. All of the above

EXERCISE

Problem

What is wireless sensor networking data acquisition?

Hands-On Projects

Project

What is the difference between the Wi-Fi NI CompactDAQ chassis and a wireless sensor node?

Case Projects

Problem

How do you add wireless sensors to your hard-wired security system?

Optional Team Case Project

Problem

How do you install window sensors for a wireless burglar alarm?

REFERENCES

[1] M. Anand, G. Ives, I. Lee. Quantifying Eavesdropping Vulnerability in Sensor Networks. Departmental papers, Department of Computer and Information Science, University of Pennsylvania, 2005.

[2] W. Xu, W. Trappe, Y. Zhang, T. Wood, in: Proceedings of the Sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ′05). ACM, 2005, pp. 48–57.

[3] J.R. Douceur, The sybil attack. in: First International Workshop on Peer-to-Peer Systems (IPTPS ′02), 2002.

[4] C. Karlof, D. Wagner, Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. First International Workshop on Sensor Network Protocols and Applications. IEEE, 2003, pp. 113–127.

[5] J. Newsome, E. Shi, D. Song, A. Perrig. The sybil attack in sensor networks: analysis & defenses. Third International Symposium on Information Processing in Sensor Networks, IPSN. IEEE, 2004, pp. 259–268.

[6] W. Xu, W. Trappe, Y. Zhang, Defending wireless sensor networks from radio interference through channel adaptation, ACM Trans. Sensor Network 4(4), 18–34.

[7] Sun Zheng, Xiao-guang Zhang, Hui Li, Anqi Li. The application of TinyOS beaconing WSN routing protocol in mine safety monitoring. International Conference on Mechtronic and Embedded Systems and Applications, MESA. IEEE, 2008, pp. 415–419.

[8] M. Healy, T. Newe, E. Lewis Security for wireless sensor networks: a review. SAS 2009 – IEEE Sensors Applications Symposium. IEEE, 2009, pp. 80–85.

[9] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, D. Culler, SPINS: Security protocols for sensor, Wireless Networks 8 (5) (2002) 521–534.
[10] C. Karlof, N. Sastry, D. Wagner. TinySec: A Lnk Layer Security Architecture for Wireless Sensor Networks. Second ACM Conference on Embedded Networked Sensor Systems. ACM, 2004, pp. 162–175.
[11] Z. Alliance, Zigbee Specification, 2013. <http://www.zigbee.org/Specifications.aspx>.
[12] H. Kupwade Patil, S.A. Szygneda, Security for Wireless Sensor Networks Using Identity-Based Cryptography, CRC Press, 2012.
[13] A.S. Wander, N. Gura, H. Eberle, V. Gupta, S.C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. Third IEEE International Conference on Pervasive Computing and Communications. IEEE, 2005, pp. 324–328.
[14] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, P. Kruus. TinyPK: securing sensor networks with public key technology. 2nd ACM Workshop on Security of ad hoc and Sensor Networks. ACM, 2004, pp. 59–64.
[15] Q. Huang, J. Cukier, H. Kobayashi, B. Liu, J. Zhang. Fast Authenticated Key Establishment Protocols for Organizing Sensor Networks. Workshop on Sensor Networks and Applications (WSNA). ACM, 2003, pp. 141–150.
[16] X. Tian, D. Wong, R. Zhu., Analysis and improvement of an authenticated key exchange protocol for sensor networks, *Commun. Lett.* (IEEE) 9 (11) (2005) 970–972.
[17] K. Ren, W. Lou, Y. Zhang, Multi-user broadcast authentication in wireless sensor networks. Proceedings of Sensor, Mesh and Ad Hoc Communications and Networks. IEEE, 2012, pp. 223–232.
[18] R.C. Merkle, Protocols for public key cryptosystems. Symposium on Security and Privacy. IEEE, 1980, pp. 122–134.
[19] M. Mitzenmacher, Compressed bloom filters. Edited by ACM. Transactions on Networking 10 (5) (2002) 604–612.
[20] D. Aranha, R. Dahab, J. López, L. Oliveira, Efficient implementation of elliptic curve cryptography in wireless sensors, Adv. Math. Commun. 4 (2) (2010) 169–187.
[21] R. Blom, An optimal class of symmetric key generation systems. Advances in Cryptology: Proceedings of Eurocrypt ′84, 1984, pp. 335–338.
[22] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, Perfectly-secure key distribution for dynamic conferences, Advances in Cryptology—Crypto ‘92, Springer-Verlag, Berlin, 1992471–486.
[23] L. Eschenauer, V.D. Gligor. A key management scheme for distributed sensor networks. Proceedings of the 9th ACM Conference on Computer and Communication Security. ACM, 2002.
[24] C. Perrig, D. Song. Random key pre-distribution schemes for sensor networks. Proceedings of the 2003 IEEE Symposium on Security and Privacy. IEEE, 2003, pp. 197–213.
[25] W. Du, J. Deng, Y. Han, P. Varshney. A pairwise key predistribution scheme for wireless sensor networks. Proceedings of the Tenth ACM Conference on Computer and Communications Security (CCS 2003). ACM, 2003, pp. 42–51.
[26] D. Liu, P. Ning. Establishing pairwise keys in distributed sensor networks, Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS 2003). ACM, 2003, pp. 52–61.
[27] J. Lee, D. Stinson, Deterministic key predistribution schemes for distributed sensor networks, Lecture Notes in Computer Science, 3357, Springer-Verlag, 2005294–307.
[28] S. Camtepe, B. Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks, *IEEE Trans. Networking* (IEEE) 15 (2) (2007) 346–358.
[29] J.N. Al-Karaki, A.E. Kamal, Routing techniques in wireless sensor networks: a survey, *Wireless Commun.* (IEEE) 11 (2004) 6–28.
[30] D. Ganesan, R. Govindan, S. Shenker, D. Estrin, Highly-resilient, energy-efficient multipath routing in wireless sensor networks, *Mobile Comput. Commun. Rev.* (ACM SIGMOBILE) 5 (4) (2001) 10–24.
[31] R.C. Shah, S. Roy, S. Jain, W. Brunette, Data MULEs: modeling a three-tier architecture for sparse sensor networks, *Sensor Network Protocols Appl.* IEEE (2003) 30–41.
[32] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, S. Singh. Exploiting heterogeneity in sensor networks. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, 2005, pp. 878–890.
[33] Y. Xiao, X. Shen, D. Du, Wireless Network Security, Springer, 2007.
[34] C. Perkins, P. Bhagwat, Highly dynamic destination sequenced distance-vector routing for mobile computers, *ACM's Comput. Commun. Rev.* (ACM) (1994) 234–244.
[35] C. Perkins, E. Royer. Ad-hoc on-demand distance vector routing. Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.
[36] D. Johnson, D. Maltz, Dynamic sourcerrouting, Mobile Computing, Kulwer Academic Press, 1996153–181
[37] Z. Haas, M. Pearlman, The performance of query control scheme for the zone routing protocol, *Trans. Networking* (IEEE) (2001) 427–438.
[38] Y. Yu, R. Govindan, D. Estrin. Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Tech Report, UCLA, 2001.
[39] Kuo-Feng Ssu, Wang Wei-Tong, Chang Wen-Chung, Detecting sybil attacks in wireless sensor networks using neighboring information, Comput. Networks (Elsevier) 53 (18) (December 2009) 3042–3056.
[40] S. Shakkottai, T.S. Rappaport, P.C. Karlsson., Cross-layer design for wireless networks, *Commun. Mag.* IEEE (2003) 74–80.
[41] L.-A. Larzon, U. Bodin, O. Schelen. Hints and notifications. Wireless Communications and Networking Conference. IEEE, 2002, pp. 635–641.
[42] T. ElBatt, A. Ephremides, Joint scheduling and power control for wireless ad hoc networks, IEEE Trans. Wireless Commun. 3 (1) (2004) 74–85.
[43] Std.802.15.4, IEEE. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs). IEEE Press, 2003.