Chapter 1

SECURITY FOR WIRELESS SENSOR NETWORKS

Sasikanth Avancha, Jeffrey Undercoffer, Anupam Joshi and John Pinkston

University of Maryland, Baltimore County
Department of Computer Science and Electrical Engineering
1000 Hilltop Circle, Baltimore, MD 21250
{savanc1, junder2, joshi, pinkston}@cs.umbc.edu

Abstract This chapter identifies the vulnerabilities associated with the operational paradigms currently employed by Wireless Sensor Networks. A survey of current WSN security research is presented. The security issues of Mobile Ad-Hoc Networks and infrastructure supported wireless networks are briefly compared and contrasted to the security concerns of Wireless Sensor Networks. A framework for implementing security in WSNs, which identifies the security measures necessary to mitigate the identified vulnerabilities is defined.

Keywords: Sensor Networks, Security, Wireless, WSN.

1. Introduction

Information technology has dramatically changed the manner in which societies protect its citizens. For example, recent media reports [14] detail the Department of Energy's (DOE) "SensorNet" project and National Oceanic and Atmospheric Administration's (NOAA) "DCNet". SensorNet consists of sensors that measure wind direction in order to forecast how urban "wind fields" might disperse fallout from a weapon of mass destruction. DCNet consists of sensors that measure gamma-radiation. Although disjoint, the sensors of each network are co-located. In their current configuration, SensorNet and DCNet transmit their data to a network of fixed and mobile relay collection stations where the data is processed. The DOE and NOAA also have similar initiatives in Manhattan. Although these prototypical "Wireless Sensor Networks" (WSNs) are limited to measurements of wind plumes and gamma radiation, it is only a matter of time until they are extended to include sensors that are able

to detect vibrations, chemicals, biological agents, explosives, footsteps, and voices.

Technology, however, is a double-edged sword. Just as nations and societies employ technology to protect themselves, their adversaries employ technology to counter and mitigate the security afforded by these new and innovative protective measures. Consequently, steps need to be taken to ensure the security of these protective technologies. In the case of a WSN, the underlying data network, the physical sensors, and the protocols used by the WSN all need to be secured.

To exemplify this double-edged sword, consider the omnipresent threat to distributed control systems (DCS) [24] and supervisory control and data acquisition systems (SCADA) [6]. These control systems are often wireless and share some similarities with WSNs. The simplest of these systems collect data associated with some metric and, based upon its value, cause some event to occur. Events include the closing or opening of railway switches, the cycling of circuit breakers, and the opening and closing of valves. Complex DCSs and SCADAs accept data from multiple sensors and using controllers, govern a wide array of devices and events in response to the measured data.

On April 23, 2000, Vitek Boden was arrested in Queensland, Australia and was eventually found guilty of computer hacking, theft and causing significant environmental damage [33]. This criminal case is of particular interest because, to date, it is the only known case where someone has employed a digital control system to deliberately and maliciously cause damage. Using a transmitter and receiver tuned to the same frequencies as the SCADA controlling Queensland's municipal water system and a laptop computer he effectively became the master controller for the municipality's water system. He then proceeded to wreak havoc upon the city's water supply by causing the release of raw sewage into local waterways and green spaces.

DCSs and SCADAs were not designed with public access in mind, consequently they lack even rudimentary security controls. Moreover, if DCSs and SCADAs were constrained by security controls, they may fail to work properly because their timing and functionality are predicated upon unfettered communications between system components.

Similarly, the security requirements of a WSN impose costly constraints and overhead due to a sensor node's limited power supply and computational resources. In a manner not unlike the Boden case, a WSN bereft of security controls will most likely suffer the same fate as did Queensland's municipal water system.

The goal of this chapter is to present a framework for implementing security in WSNs. Existing WSN research, which is minimal, is identified and is presented. The threats to a WSN are identified, the operational paradigms utilized by WSNs and their associated vulnerabilities are presented. The se-

curity models applicable to other types of wireless networks, (e.g.: 802.11b or Bluetooth-based wireless Internet and MANETs), are compared and contrasted to the security models that are specified for WSNs. The WSN security state-of-the-art is detailed, identifying where the state-of-the-art falls short of the ideal. Ideas are presented for moving the state-of-the-art closer to the ideal, recommendations are made, and conclusions are drawn.

2. Threats to a WSN

There are many vulnerabilities and threats to a WSN. They include outages due to equipment breakdown and power failures, non-deliberate damage from environmental factors, physical tampering, and information gathering. We have identified the following threats to a WSN:

Passive Information Gathering: If communications between sensors, or between sensors and intermediate nodes or collection points are in the clear, then an intruder with an appropriately powerful receiver and well designed antenna can passively pick off the data stream.

Subversion of a Node: If a sensor node is captured it may be tampered with, electronically interrogated and perhaps compromised. Once compromised, the sensor node may disclose its cryptographic keying material and access to higher levels of communication and sensor functionality may be available to the attacker. Secure sensor nodes, therefore, must be designed to be tamper proof and should react to tampering in a *fail complete manner* where cryptographic keys and program memory are erased. Moreover, the secure sensor needs to be designed so that its emanations do not cause sensitive information to leak from the sensor.

False Node: An intruder might "add" a node to a system and feed false data or block the passage of true data. Typically, a false node is a computationally robust device that impersonates a sensor node.

While such problems with malicious hosts have been studied in distributed systems, as well as ad-hoc networking, the solutions proposed there (group key agreements, quorums and per hop authentication) are in general too computationally demanding to work for sensors.

Node Malfunction: A node in a WSN may malfunction and generate inaccurate or false data. Moreover, if the node serves as an intermediary, forwarding data on behalf of other nodes, it may drop or garble packets in transit. Detecting and culling these nodes from the WSN becomes an issue.

Node Outage: If a node serves as an intermediary or collection and aggregation point, what happens if the node stops functioning? The protocols employed by the WSN need to be robust enough mitigate the effects of outages by providing alternate routes.

Message Corruption: Attacks against the integrity of a message occur when an intruder inserts themselves between the source and destination and modify the contents of a message.

Denial of Service: A denial of service attack on a WSN may take several forms. Such an attack may consist of a jamming the radio link or could it could exhaust resources or misroute data. Karlof and Wagner [21] identify several DoS attacks including: "Black Hole", "Resource Exhaustion", "Sinkholes", "Induced Routing Loops", "Wormholes", and "Flooding" that are directed against the routing protocol employed by the WSN..

Traffic Analysis: Although communications might be encrypted, an analysis of cause and effect, communications patterns and sensor activity might reveal enough information to enable an adversary to defeat or subvert the mission of WSN. Addressing and routing information transmitted in the clear often contributes to traffic analysis. We further address traffic analysis in the following subsection.

Traffic analysis is an issue that has occasionally attracted the attention of authors writing about security in networks [11, 12, 31]. Traffic analysis is the term used for the process of inferring information about the communications of an encrypted target network. Although unable to read the encrypted message contents, the analyst examines the externals - which station is sending, to whom messages are sent, and the patterns of activity. Sometimes, the identities of the correspondents are contained in unencrypted message headers, while at other times they can be inferred by direction finding techniques on radio signals. The order of battle can be deduced using traffic analysis - i.e., which station is the command headquarters, which ones are at intermediate levels, and which are at the lowest echelons. Patterns of activity can also be used to deduce the target's alert status - e.g., routine, heightened, attack imminent.

Classically, traffic analysis is countered by communication systems employing traffic flow security. In this mode, the system transmits an encrypted stream continuously, encrypting idle messages when there is no valid traffic to be sent. In this way, the unauthorized listener can not tell when the parties are actually communicating and when they are not, and is thus unable to make traffic analysis deductions. With radio links in years past, traffic flow security was often employed. Doing so did not add significant expense - it occupied a radio frequency continuously, but radio spectrum was not at a premium, nor was electric power for the transmitter a concern.

With WSNs having limited-energy nodes, the practicality of traffic flow security becomes quite problematic. Now, the cost of sending dummy traffic results directly in a reduction in the lifetime of the node. The energy required for communications is typically the dominant factor in battery lifetime of a node, so the lifetime is reduced by the ratio of dummy traffic to real traffic,

which must be substantially greater than 1, to provide any useful traffic flow security protection. This will be intolerable in virtually all cases; consequently traffic flow security for WSNs is in need of further research. The effects of traffic analysis may be partially mitigated by encrypting the message header that contains addressing information, however further research is needed.

3. WSN Operational Paradigms and Their Corresponding Vulnerabilities

We categorize WSNs according to its operational paradigm. Some models of operation are simple; the sensor takes some measurement and blindly transmits the data. Other operational models are complex and include algorithms for data aggregation and data processing. In order to discuss security measures for a WSN sensibly, one must know the threats that must be defensed, and equally important, those that need not be provided for. It makes no sense to even attempt to design protection against an all-powerful adversary, or even against an adversary who gets the last move in a spy-vs-spy, move-countermove game. One must select a model of the adversary's capabilities and work to that.

A security architecture will depend on our assumptions about the integrity of the hardware and software of the base stations and outstation nodes, and our assumptions about the capability of an adversary to eavesdrop, intrude on the network, or obtain physical access to a node.

What do we need to protect the WSN? Is it the content of the data being communicated (confidentiality)? Do we need to ensure the correctness of this data (data integrity)? Does the system need to be robust and survivable when confronted by a sophisticated adversary? What do we assume about the adversary's capability to physically capture and subvert one of the deployed nodes, and to extract sensitive information such as cryptographic keys from it? The answers to these questions will guide the designer to incorporate the right capabilities.

Typically, to ensure the desired level of system operation, one must have confidence in the correctness and completeness of the data that is collected and forwarded to the point(s) where decisions are made and responses initiated. One may or may not care if anyone else knows that the network is there, or can understand the traffic.

The following briefly describes the operational paradigms that a WSN may use. In each case, we assume the presence of a base station, or controller.

Simple Collection and Transmittal. The sensors take periodic measurements and transmit the associated data directly to the collection point. Transmission occurs either immediately following data collection or is scheduled at some periodic interval. In this paradigm each node is only concerned with its trans-

mission to the base station, which is assumed to be within range. Thus, any notion of routing or co-operation among nodes is absent from this paradigm.

This operational paradigm is vulnerable to attacks directed against the Link Layer. Denial-of-service attacks include *jamming* the radio frequency and *collision induction*. It is also vulnerable to *spoofing* attacks in which a counterfeit data source broadcasts spurious information. If the data is considered to be sensitive and it is not encrypted, then a loss of confidentiality may occur if someone passively monitors the transmissions emanating from the WSN. This paradigm and all the following are also susceptible to physical attacks - capture of a node and subsequent subversion. Such threats are countered by tamper resistant technologies, which may transmit an alert and/or self destruct when tampering is detected. Discussion of these techniques is beyond the scope of this chapter. Replay attacks in which an adversary transmits old and/or false data to nodes in the WSN can also be mounted on the six paradigms discussed here.

Forwarding. Sensors collect and transmit data to one or more neighboring sensors that lie on a path to the controller. In turn, the intermediate sensors forward the data to the collection point or to additional neighbors. Regardless of the length of the path, the data eventually reaches the collection point. Unlike the first paradigm, co-operation among nodes in "routing" the data to the base station is part of this paradigm. That is, a node that receives data intended for the base station attempts to transmit the same toward the latter, instead of throwing the data away.

In addition to being vulnerable to the attacks identified under the Simple Collection and Transmittal paradigm, this method is also vulnerable to *Black Hole*, *Data Corruption* and *Resource Exhaustion* attacks. In a Black Hole attack, the sensor node that is responsible for forwarding the data drops packets instead of forwarding them. A Data Corruption attack occurs when the intermediate node modifies transient data prior to forwarding it. These attacks require that the node be subverted or that a foreign, malicious node be successfully inserted into the network. A Resource Exhaustion attack occurs when an attacker maliciously transmits an inordinate amount of data to be forwarded, consequently causing the intermediate node(s) to exhaust their power supply.

Receive and Process Commands. In this paradigm, sensors receive commands from a controller, either directly or via forwarding, and configure or re-configure themselves based on the commands. This ability to process commands is in addition to that of transmitting unsolicited data to the controller and helps in controlling the amount of data handled by the WSN. In this model, the communication paradigm changes from being exclusively many-to-one to now include one-to-many communication which means that whereas in the former, the data transmitted was intended *only* for the base station, in the latter, the data

(i.e., command) is applicable to one or more sensor nodes. Commands may be broadcast to the entire WSN or may be unicast to a single sensor. If unicast messaging is employed, then some form of addressing of each individual node needs to be employed. However, no guarantees on the unicast message actually reaching the intended recipient can be given, because none of the nodes in the WSN may be aware of either route(s) to the recipient or the topology of the WSN.

In addition to being vulnerable to all of the aforementioned attacks, the Receive and Process Commands paradigm is also vulnerable to attacks where an adversary impersonates the controller and issues spurious commands.

Self-Organization. Upon deployment, the WSN self organizes, and a central controller(s) learns the network topology. Knowledge of the topology may remain at the controller (e.g. base station) or it may be shared, in whole or in part, with the nodes of the WSN. This paradigm may include the use of more powerful sensors that serve as cluster heads for small coalitions within the WSN.

This paradigm requires a strong notion of routing, therefore, in addition to being vulnerable to all of the previously listed attacks, this paradigm is vulnerable to attacks against the routing protocol. These attacks include *Induced Routing Loops, Sinkholes, Wormholes* and *HELLO Flooding*. They are presented and discussed in detail in [21].

Data Aggregation. Nodes in the WSN aggregate data from downstream nodes, incorporating their own data with the incoming data. The composite data is then forwarded to a collection point.

This paradigm is particularly vulnerable to replay attacks because the *authentication* of its downstream peers becomes an issue. In the previous paradigms, the authentication of the sensor node was left to the controller, which is not an issue because controllers are robust and considerably more powerful than the sensor nodes. In this paradigm, each sensor node that utilizes data from another sensor node now can not just forward the data as received, and therefore must ensure that the data is provided by an authorized member of the WSN.

Optimization: Flexibility and Adaptation. Predicated upon their own measurements and upon the values of incoming data, this paradigm requires that the sensors in the WSN make decisions. For example, a decision may be whether to perform a calculation, or if the cost is less, acquire the needed value from a peer, provided that the peer has the value and that knowledge is known in advance by the requester.

This operational paradigm shares the same security concerns and issues as does the Data Aggregation paradigm.

4. Toward a Generic Security Model for WSN

The previous section illustrated that as functional and communication complexity increases, so do the vulnerabilities. The following section examines the manner in which these vulnerabilities are mitigated by the application of security measures. We show that as the communication model becomes more complex, the security measures become non-trivial and require increasingly sophisticated solutions. We conclude this section by proposing a generic security model that is applicable across all WSN operational paradigms.

The overlap between security issues and their associated solutions in infrastructure supported wireless networks and mobile ad-hoc networks (MANETs) is minimal and usually quite different from those in WSNs. Before discussing WSN security models in detail, we briefly describe the security models applied to infrastructure-supported wireless networks and MANETs noting where they are applicable to WSNs.

Infrastructure supported wireless networks, e.g., wireless LANs, use a centralized, point-to-point communication model and are user-centric – users with wireless devices access the wired infrastructure via a *wireless base station*. The base station is the sole interface between users and the wired infrastructure and is responsible for security at the link layer, although additional security measures at the network and application layer may be applied. The primary security measure implemented in the base station is link-level encryption. User authentication and access control may be applied at the base station or it may be applied at points further upstream from the base station. Regardless, these functions are invoked only when a user attempts to access the wired infrastructure via the base station. Due to the user-centric nature of these networks, authentication and access control are of greater concern than confidentiality. Additionally, the point-to-point communication model ensures that end-to-end security is enabled. The centralized nature of these networks makes them vulnerable to attacks such as spoofing, passive data gathering and denial-of service, most of which are directed at the base station. When encryption is implemented, the goal of the attacker becomes the compromise of the encryption key. The travails of the Wired Equivalent Privacy (WEP) protocol in 802.11b for wireless LANs [13] are well-documented and serve to highlight the security problems of these networks.

In contrast to wireless networks, MANETs employ a distributed, multi-hop, node-centric communication model. In a MANET, users control their wireless devices, however the device itself has some degree of autonomy. The autonomy is best illustrated by a device's choosing of the most appropriate set of neighboring devices to contact based on user requirements. Thus, communication in MANETs is node-centric, rather than user-centric. We may immediately observe that device authentication and data confidentiality are much

more important than access control, which may not be relevant in certain situations. Unlike infrastructure-supported wireless networks, the security problem confronted by a MANET is to mitigate the actions of *malicious* users who may attempt to disrupt communication in the network. Thus, security protocols in MANETs employ mechanisms such as certificates to authenticate users and encrypt data using symmetric or asymmetric algorithms. Due to the fact that MANETs allow multi-hop communications, most attacks are directed against the protocols that route data between intermediate nodes on the path from source to destination. Thus, network-level security is the focus of attention in security research related to MANETs. Protocols and methods designed to address this issue include SEAD [16], Ariadne [17], security enhancements in AODV [5], secure position aided routing [8], secure ad hoc routing [29] and an on-demand secure routing protocol resilient to Byzantine failures [3]. End-to-end security is a non-trivial problem in MANETs because security protocols must rely on intermediate nodes which, depending upon their individual capabilities, may not contain all of the mechanisms required by the protocols.

Let us now explore the security requirements, features, problems and possible solutions for each of the six WSN operational paradigms described in Section 1.1.

4.1 Simple Collection and Transmittal

In this operational paradigm, sensor nodes sense and transmit the associated data to the collection point (e.g.: base station; controller). This paradigm, therefore, uses a centralized, point-to-point, single-hop communication model similar to that in infrastructure-supported wireless networks. The primary security requirements of this paradigm are data access, authentication and data confidentiality. Confidentiality can be ensured by the use of data encryption. Symmetric key encryption methods, such as DES [28], are suitable for this paradigm. Authentication is implicitly ensured by the use of pre-deployed keys that are shared between, and unique to, the collection point and each individual sensor node [2]. Each node uses its key to encrypt data before transmission; the collection point decrypts the data using the shared key corresponding to that node.

Spread spectrum communications may be used to offset efforts to jam the frequency band. Error correcting codes offer some relief to collision inductions. End-to-end security is trivially enabled by encrypting the data frame; because each sensor node can communicate directly with the collection point.

4.2 Forwarding

A straightforward solution to the "radio range" problem presented in the Simple Collection and Transmittal paradigm is to allow sensor nodes far away

from the collection point to transmit data to neighboring sensor nodes, which in turn *forward* the data toward the collection point. The forwarding process may span multiple sensor nodes on the path between the source node and the collection point. Thus, this paradigm uses a centralized, multi-hop communication model. The overarching requirement to minimize energy consumption complicates possible security mechanisms.

We observe that a single shared key between the collection point and a sensor node is no longer sufficient to ensure reliable transmission of a node's data to the collection point. Why? Consider, for example, the case of a sensor node positioned two hops away from the collection point. It shares a key with the collection point and encrypts its data using this key. The sender requires one of its neighbors to forward the encrypted data to the collection point. However, due to random deployment, it is quite unlikely that the originating node is aware of the identity of any of its neighbors. Therefore, it must resort to limited broadcasting. Now, the encrypted data arrives at some intermediate node one hop away from the collection point. The question is, how does this node know that the message contains valid data and is not part of an attack mounted by some malicious node? On a similar note, how can the originating sensor node be confident that the intermediate sensor node is not malicious and will actually forward the encrypted data to the collection point instead of dropping it?

To resolve these issues, Avancha et al. [2] introduce a system that utilizes pre-built headers encrypted under the intermediate node's key. At the origin, the entire frame intended for the controller is encrypted under the senders key and inserted into another frame that is prepended with the pre-built header and broadcast. When the intermediate node receives the broadcast, it strips off the prepended header and re-broadcasts the frame. It is then received by the controller and decrypted. We note that this solution possesses limited scalability in terms of number of hops; as the number of hops increases, the number of pre-built headers to prepend also increases leading to increased message size.

Avancha et al. [2] also present a mechanism to mitigate the effects of a *Black Hole* and *Data Corruption* attacks, which could possibly be effected by a compromised intermediate node. The mechanism is in the form of an algorithm that employs counters and timers. It tracks the absence of expected data from each sensor, quantifies the amount of corrupted data received at the controller from each sensor, and compares those values to acceptable statistical norms. If the controller determines a sensor node to be aberrant, it is culled from the WSN.

4.3 Receive and Process Commands

The paradigms discussed in sections 1.4.1 and 1.4.2 above described a "many-to-one" communication model, designed exclusively for *unsolicited* data transmissions. (We define unsolicited data transmissions as those that emanate from sensor nodes without any external stimulus, such as a command, and are directed toward the controller.) As is well-known, data transmission is the most expensive of all operations in a WSN. Thus, unsolicited data transmissions, especially if they are unnecessary (e.g.: 100 sensor nodes reporting directly to a controller that the temperature in the region is 90 F), may reduce the lifetime of the WSN significantly. If transmissions could be regulated appropriately, then the lifetime of the WSN could be increased without affecting the quality of information reaching the controller. This calls for the use of a "one-to-many" communication model, where the controller transmits commands to the sensor nodes.

Consider, for example, a group of sensor nodes that are deployed in order to monitor temperature. Upon deployment, all nodes begin operating in the *idle mode*, which is a low-power mode. The controller broadcasts a *wakeup* command to a set of sensor nodes, which react to this command by transitioning to the *active* state. Subsequently, the controller broadcasts a *getdata* command, which solicits data from the sensor nodes. Finally, the controller instructs the sensor nodes to *idle* and the cycle repeats periodically. Thus, the combination of the *one-to-many* and *many-to-one* communication models is more energy-efficient than simply using the latter for unsolicited data transmission.

As expected, the cost of improved energy efficiency in this paradigm requires a more complicated security model. In the previous paradigm, the main security issue was to build *trust* among sensor nodes so that they could cooperate in data transmission to the collection point. This issue is now extended to the controller. How does a sensor node authenticate the command it received as being broadcast by the controller? This issue is further complicated by the use of multi-hop communications. A sensor node that is two hops away from the controller must depend upon some set of "intermediate neighbors". How does such a sensor node verify the integrity of the message it receives from its neighbors? How can it satisfy itself that the message was not tampered with by an intermediate node along the path from the controller?

The first issue can be addressed by the use of *broadcast authentication* as discussed in the SPINS project [30] or by the use of shared secrets between the controller and the individual sensor nodes. The second issue can be resolved by distributing encrypted identities of sensor nodes within radio range of the controller among those sensor nodes that are beyond radio range of the controller by using pre-built headers, presented in the Section 1.4.2.

4.4 Self-Organization

The operating paradigm presented in the Section 1.4.3 describes secure bi-directional communications between the controller and the sensor nodes. This paradigm in this section extends the bi-directional communication model by introducing the concept of *self-organization*. The self-organization paradigm requires that the WSN achieve organizational structure without human intervention. It consists of three primary tasks: *node discovery*, *route establishment* and *topology maintenance*. The accomplishment of these three tasks leads to the formation of a true WSN. While the previous paradigms used a centralized communication model, this and the following paradigms seek to employ a combination of centralized and distributed communication in order to allow the WSN to perform as efficiently and securely as possible.

In the node discovery process, the controller or a sensor node broadcasts a discovery message, e.g., a *HELLO* message. In response to this message, nodes unicast a message indicating their presence in the proximity of the broadcaster, e.g., a *HELLO-REPLY* message. This message sequence is sufficient for establishing a secure single-hop WSN. For a multi-hop WSN, this sequence must be augmented by the encrypted node identities or some other mechanism. An important point to note is that node discovery itself must be performed in a secure, authenticated manner to mitigate the effects of *traffic analysis* and replay attacks as much as possible.

Following node discovery, routes between the sensor nodes and the controller must be established. In order to ensure continuous connectivity, multiple routes between a pair of nodes may be established. An important question that arises in this context is: Should security in the WSN be end-to-end or be restricted to pair wise security between nodes? As discussed Karlof and Wagner in [21], most existing routing protocols for WSNs are vulnerable to a host of attacks including flooding , wormhole [18], sinkhole and Sybil [9] attacks. Consequently, the routing protocol is extremely important and needs to be secure.

In order to protect against attacks to the routing protocol, all routing information that is distributed throughout the WSN needs to be encrypted, be protected by an anti-replay mechanism and its source needs to be authenticated. Sensors are so resource constrained that they cannot maintain key tables containing all of their neighbors keys, nor can they use asymmetric encryption. Currently, a key shared across the WSN is the most viable solution to protecting routing information (i.e.: source and destination addresses). Moreover, as is the case whenever a sensor contains cryptographic keying material, the physical sensor needs to be configured so that tampering will erase the keys and render the sensor inoperable. The sensor also needs to be fabricated so that keys are not leaked via electronic emanations.

Topology maintenance in a WSN is unlike that in any other wireless network. In WSNs, nodes are stationary. Therefore, the topology, once established, usually does not change. However, as nodes perform their assigned tasks they deplete and eventually exhaust their energy store, causing them to die. The WSN may be "refreshed" by the periodic addition of new nodes to the WSN. The addition of new nodes to the existing network also implies additional security concerns. Both the node discovery and route establishment algorithms need to be re-run, and whether the algorithms are centralized or distributed is cause for additional concerns. If the WSN is centralized and shared secrets are used for encryption, then the keys of the new nodes must be deployed on the controller using either a secure key distribution algorithms or programmed with the same key schedule as in SPINS. If the WSN is distributed, then key management procedures must be invoked to ensure that the new nodes possess the relevant encryption keys. This is accomplished by loading the controller and the new sensor node with the shared key.

4.5 Data Aggregation

So far, our discussion of WSNs and security has assumed that all sensor nodes transmit their data directly to the base station in either a solicited or unsolicited manner. Under this assumption, the sensors are not dependent upon the integrity or authenticity of the data. This results in the hundreds or perhaps thousands of independent *data streams*. An important problem in WSNs is to control these data streams so that unnecessary data transmissions can be eliminated and the collection point can be prevented from becoming a bottleneck.

A substantial body of research on wireless sensor networks is devoted to researching the problem of controlling data streams [19, 22, 27, 26, 34]. The prevailing solution to this problem is to *aggregate* or *fuse* data within the WSN and transmitting an aggregate of the data to the controller. The idea therefore, is to allow a sensor node to transmit its data to its neighbors, or some subset thereof. In turn, some algorithm controls which node will combine the data received from its neighbors and forward it toward the controller. This data aggregation process results in a substantial energy savings in the WSN. Typical aggregation operations include MAX, MIN, AVG, SUM and many other well-known database management techniques.

The following exemplifies the averaging (AVG) methodology. Consider the case of 100 sensor nodes deployed to measure temperature and transmit the collected data to the controller. Without data aggregation, 100 temperature readings must be transmitted, possibly multiple times, to the collection point. With data aggregation, if 1 in 10 nodes performs the AVG operation on data received from its immediate neighbors, the total number of transmissions to the collection point is reduced from 100 to only 10. The problem of control-

ling data streams is now reduced to choosing the most appropriate *aggregation points*, i.e., the subset of nodes to perform aggregation. A number of solutions to this problem are discussed in literature, including [22, 20, 25].

However, the implicit assumption is that the sensor nodes trust each other so that any pair of nodes can exchange data and that a node can incorporate the incoming data with its own. The problem with this assumption is that it does not take into account the very real possibility that an adversary may have deployed malicious data sources in the WSN for nefarious purposes. For example, a malicious node may endeavor to have itself elected as an aggregation point and then throw away all of the data that it receives from its neighbors, or even worse, transmit corrupt or fictitious data to upstream neighbors. Again, this operational paradigm requires an even more complex set of security controls. At first glance, the problem may seem trivial, given that secure routes would have been established during the self-organization process described in Section 1.4.4. However, the secure routes were established without considering data aggregation and the choice of aggregation points. Thus, if end-to-end security were established between the controller and each sensor node in the WSN, then data aggregation is not possible. On the other hand, point-to-point security allows complete flexibility in the choice of aggregation points, but is not a scalable solution because each node would have to keep cryptographic information for all other nodes in the WSN. Thus, a simple solution to the *secure data aggregation* problem is the use of a key that is common to the WSN. The intrinsic weakness in this approach is the use of a common key, if it is somehow compromised then so is the entire WSN. In [15] Hu and Evans discuss a time delayed protocol to securely aggregate data in wireless networks.

4.6 Optimization: Flexibility and Adaptability

The WSN paradigms thus far considered focus on the data gathering and reporting functions of a WSN. The nodes in the WSN are not concerned with the *semantics* of the data they have obtained through the sensing task. The sole concern is that it must be transmitted elsewhere, possibly for further analysis. To this end, all the nodes execute a fixed set of protocols, one protocol per the link, network and security layers, irrespective of the environmental and security conditions affecting the WSN. Inflexibility in the choice of protocols and the inability to adapt to changing conditions could render a WSN inoperable or cause it to function sub-optimally. Here, optimality encompasses both energy usage and security. Consider, for example, a WSN that uses a centralized, point-to-point communication model as described in Section 1.4.1. Given the cost of transmitting at full power, the utility of this type of WSN is limited to an application that requires surveillance and monitoring in a relatively small area, such as a single room in a building. By implication, the same WSN cannot

be used for monitoring purposes over large areas, such as a bridge or stadium. On the other hand, an agile WSN, one that is capable of functioning in either a centralized mode or a distributed mode can be deployed without the constraints imposed by the area of coverage. Furthermore, if the WSN is provided with multiple combinations of link, network, security and aggregation protocols, then it can dynamically choose a particular combination based upon existing environmental and security conditions. Finally, permitting the WSN to use the data that it collects from the environment, to make on-the-fly decisions regarding protocol execution will enable it to self-optimize in terms of energy usage and security profile. For example, if the WSN senses harmful chemicals in the environment along with a sudden increase in radio noise levels (indicating a Denial of Service attempt by some adversary), it can take appropriate countermeasures by re-routing around the jammed areas. Thus, the WSN exhibits both flexibility (in terms of protocol execution) and adaptability (in terms of protecting itself), which are most desirable attributes of a WSN.

4.7 A Generic WSN Security Model

We now describe the principal components of an ideal, generic security model for wireless sensor networks. Some components of this model, such as communication security and key management, have been and continue to be topics of active research; others, such as data aggregation and self-healing, have yet to receive a considerable amount of attention.

Communication model: Hybrid communication employing both centralized and distributed models; the centralized model is used when one or more powerful nodes exist, around which less-powerful sensor nodes can *cluster* and the distributed model is employed when no powerful nodes exist. These models can be used together at the same time to form a hierarchical WSN – the centralized model to first form clusters and the distributed model for *inter-cluster* communication.

Communication security: As was the case with the communication model, the mechanisms to secure communication between nodes are also deployed in a hybrid manner. In the case where more powerful nodes exist and clusters can be formed, end-to-end communication security between the designated *clusterhead* and each individual sensor node in the cluster should be used. Subsequently, inter-cluster communication security, i.e., communication between clusterheads, should be pair wise. In the absence of more powerful nodes, as the WSN is formed in a distributed manner, it is appropriate to employ pair wise security, but only for a fixed number of pairs. This is because pair wise security is not scalable as the number of nodes in the WSN increases. Thus, pure pair wise security is not feasible in the ideal security model.

Key management: Due to the fact that most sensor nodes in a WSN have limited amount of energy, public-key cryptographic mechanisms, which are expensive in terms of energy consumption, are not suitable to WSNs. Private-key cryptography, on the hand, is quite applicable to WSNs due to its low energy requirements. However, in a hybrid WSN that consists of nodes of varying capabilities and resources, it is feasible to employ both public-key and private key mechanisms for security. Thus, intra-cluster communications are secured by private-key cryptography and inter-cluster communications via public-key cryptography. An additional problem in WSNs, although not unique to them, is key distribution. The principal mechanisms to solve this problem are pre-deployed keys (i.e., offline key distribution), group keying and arbitrated keying [7]. In the ideal model, all three mechanisms are interchangeably used predicated upon the exact composition of the WSN. If the WSN consists of clusters, then either arbitrated or group keying is appropriate. A flat topology in the WSN calls for pre-deployed keys so that either pair wise or end-to-end security may be employed by the nodes.

Data aggregation: In the ideal security model, data aggregation can be performed as often as required and in a manner that conforms to the security requirements. Additionally, based on chosen communication model, different aggregation algorithms can be executed at different points during the lifetime of the WSN. For example, at start up, the WSN may have a single controller and many simple sensor nodes. In this case, data aggregation may be performed by the controller itself. This calls for a data aggregation algorithm that is suitable to the centralized communication model. As additional nodes are added, the WSN may switch to the distributed communication model; data aggregation should be performed either by multiple designated or elected nodes based on the security mechanism, i.e., end-to-end or pair wise, that is in effect at that point in time.

Self-healing: Security models for WSNs face problems not only in the form of external attacks on the network, but also in the form of breakdowns due to node failure, especially due to energy exhaustion. The ideal model is able to withstand the various types of attacks detailed in Section 1.3, employing both passive and active mechanisms. Passive mechanisms include data encryption and node authentication, while active mechanisms include key revocation and removing offending nodes from the WSN. In the ideal model, security countermeasures exist at every layer: spread-spectrum techniques at the link-layer, encryption at the network and application layers, authentication at the application layer and aberrant behavior detection at the network and application layers. It may not be feasible or required to activate all these mechanisms at the same time, rather they are invoked in an application-specific and environment-specific manner. Node failure causes route breakdown and may cause failure

of end-to-end or pair wise communication security, possibly leading to network partition. The ideal security model consists of mechanisms to monitor and track the health of all nodes in the network, thereby enabling quick (re)-establishment of secure routes around nodes that provide indications of imminent failure. Based on the communication model, this may require invoking procedures to distribute keys to neighboring nodes as required.

5. WSN Security: State-of-the-Art

Security aspects of wireless sensor networks have received little attention compared to other aspects. Key management in sensor networks has been dealt with to a certain extent, but research and development of security architectures has been less extensive. In this section, we present a brief overview of various key management protocols and security architectures for WSNs, including our contributions.

5.1 Key Management

Basagni et al. [4] present a key management scheme for pebblenets, defined as large ad hoc networks consisting of nodes of limited size called pebbles. The key management scheme uses symmetric cryptography. In this scheme, each pebble is equipped with a group identity key which enables it to participate in key management. Data traffic is secured using a global key shared by all nodes, called the Traffic Encryption Key (TEK). TEKs are periodically refreshed. TEK generation and distribution requires selection of a key manager to perform these tasks. The goal of this work, therefore, is to select a key manager. The protocol designed to achieve this goal consists of two phases. First, pebbles organize into a cluster with a clusterhead. The clusterheads subsequently organize into a backbone. Finally, a fraction of the clusterheads of the backbone is selected among which a pebble is chosen as the new key manager.

Carman et al. [7] have conducted a detailed study of various keying protocols applicable to distributed sensor networks. They classify these protocols under pre-deployed keying, arbitrated protocols, self-enforcing autonomous keying protocols and hybrid approaches. The authors also present detailed comparisons between various keying protocols in terms of energy consumption.

Eschenauer and Gligor [10] present a key-management scheme for sensor network security and operation. The scheme includes selective distribution and revocation of keys to sensor nodes as well as node re-keying without substantial computation and communication capabilities. It relies on probabilistic key sharing among the nodes of a random graph and uses simple protocols for shared-key discovery and path-key establishment, and for key revocation, re-

keying and incremental addition of nodes. The security and network connectivity characteristics supported by the key-management scheme are discussed.

5.2 Security Architectures

The Security Protocols for Sensor Networks (SPINS) project [30] consists of two main threads of work: an encryption protocol for SmartDust motes called Secure Network Encryption Protocol (SNEP) and a broadcast authentication protocol called micro-Timed, Efficient, Streaming, Loss-tolerant Authentication (μTESLA). In SPINS, each sensor node shares a unique master key with the base station. Other keys required by the SNEP and the μTESLA protocols are derived from this master key. SNEP is based on Cipher Block Chaining implemented in the Counter mode (CBC-CTR), with the assumption that the initial value of the counter in the sender and receiver is the same. Thus, the sender increments the counter after sending an encrypted message and the receiver after receiving and decrypting it. To achieve authenticated broadcasts, μTESLA uses a time-released key chain. The basic idea resolves around the unidirectionality of one-way functions. There are two requirements for correct functioning of this protocol: (i) the owner of the key release schedule has to have enough storage for all the keys in the key chain (ii) every node in the network has to at least be loosely time synchronized, i.e. with minor drifts. The time-released key chain ensures that messages can be authenticated only after receiving the appropriate key in the correct time slot.

Karlof and Wagner [21] consider routing security in sensor networks. This work proposes security goals for sensor networks, presents classes of attacks and analyzes the security of well-known sensor network routing protocols and energy-conserving topology maintenance algorithms. The authors conclude that all the protocols and algorithms are insecure and suggest potential countermeasures. The attacks discussed in this work include bogus routing information, selective forwarding, sinkholes, Sybil, wormholes and HELLO flooding.

Communication security in wireless sensor networks is addressed in [32]. The approach in this work is to classify the types of data that typically exist in sensor networks and to identify possible communication security threats according to that classification. The authors propose a scheme in which each type of data is secured by a corresponding security mechanism. This multi-tiered security architecture where each mechanism has different resource requirements, is expected to enable efficient resource management.

Law et al. [23] discuss security aspects of the EYES project, which is concerned with self-organizing, collaborative, energy-efficient sensor networks. This work contains three contributions. The first is a survey that discusses the dominant issues of energy-security trade-off in network protocol and key management design. This survey is used to chart future research directions for

the security framework in EYES. Second, the authors propose an assessment framework based on a system profile that enables application classification. Third, some well-known cryptographic algorithms for typical sensor nodes are benchmarked. This work also investigates resource requirements of symmetric key algorithms RC5 and TEA.

Our contributions to WSN security include two security architectures that employ centralized and distributed security models respectively. The first architecture [2], useful for applications such as perimeter protection, assumes the existence of a single controller in the WSN. Keys are pre-deployed on all sensor nodes; each sensor node shares a unique key with the controller. We use DES with 64-bit keys for encryption. The controller is responsible for secure node discovery, route establishment and topology maintenance. A unique feature of this architecture is its invulnerability to traffic analysis due to end-to-end encryption. Additionally, the architecture consists of a network repair protocol that detects and eliminates from the network, aberrant nodes – those that have either been compromised by some adversary or have exhausted their energy. The second architecture [1], employs a clustering approach to form a secure WSN. This architecture assumes the existence of a few powerful nodes around which other sensor nodes cluster. The clustering protocol is a modification of the centralized protocol used in the first architecture. The powerful nodes, called clusterheads, are responsible for secure self-organization within the cluster. After the formation of clusters, the clusterheads form a chain in order to be able to exchange and aggregate data generated in individual clusters. The controller may be either static or mobile and receives the final aggregated data from one of the clusterheads. Intra-cluster security is end-to-end, inter-cluster security is pair wise in this architecture.

6. State-of-the-Art to Ideal

In this section, we discuss the research effort required to bridge the gap between the state-of-the-art in WSN security and the ideal, in terms of the principal components discussed in Section 1.4.7.

Let us consider the current security architectures for WSNs and the requirements of communication models with integrated communication security. It is evident that the principal focus of WSN security has been on centralized approaches; there is a need to develop distributed approaches and combine them with the centralized approaches to design robust hybrid models for communication security. Our work [1] attempts to make progress in this direction. The use of hybrid models for communication security will enable easier integration of data aggregation and key management algorithms. This will also ensure flexibility and adaptability of the WSN; self-healing mechanisms and the ability to react to changing conditions can also be easily integrated.

Both centralized and distributed key management techniques for WSNs have been discussed in literature. Research efforts directed toward this problem have shown that key distribution in WSNs can be energy-efficient and secure under certain conditions. However, we observe that distributed key management techniques, as discussed in literature, are completely independent of any security architecture. For example, the work on secure pebblenets [4] is mainly concerned with choosing a key manager in every round, but does not address the issues involved in using the key for encryption, authentication or other security functions. On the other hand, the SPINS project [30] and our efforts [2, 1] assume pre-deployed keying in the entire security architecture. The ideal security model will consist of a combination of robust, energy-efficient, secure key distribution mechanisms with well-defined, comprehensive security architectures.

A similar situation is observed when security architectures and data aggregation algorithms are considered. Current security architectures do not really consider the issue of integrating data aggregation algorithms; rather they assume that designated nodes, such as the controller or clusterheads, will perform the required aggregation function(s). On the other hand, data aggregation algorithms assume complete and unhindered co-operation among all sensor nodes in the WSN as far as performing the aggregation function(s) is concerned. This assumption is non-trivial; a security protocol that supports such a co-operation model will not be scalable because *pair wise* communication security is required across the WSN. Thus, integration of well-known, energy-efficient data aggregation algorithms [22, 20, 25] with robust security architectures is essential in designing the ideal security model.

Flexibility, adaptability and self-healing mechanisms are essential to the functioning of a WSN and optimal resource use during its lifetime. None of the existing security architectures use the data associated with sensed environmental conditions to help detect the beginnings of attacks or of aberrant behavior by nodes. This reduces the ability of the WSN to protect itself from attacks mounted within and outside the network. In fact, detecting and preventing attacks from within, i.e., attacks mounted by compromised nodes, is a harder problem than preventing external attacks such as jamming. Thus, the move toward the ideal security model calls for the design and development of compact, lightweight mechanisms to capture and *reason* over data describing environmental and security conditions.

7. Conclusions

Improvements in wireless networking and micro-electro-mechanical systems (MEMS) are contributing to the formation of a new computing domain – distributed sensor networks. These ad-hoc networks of small, fully pro-

grammable sensors will be used in a variety of applications: on the battlefield, as medical devices, in equipment maintenance and in perimeter security systems.

Unless security is considered during the design of the physical sensor, its protocols and operational models, sensor networks will remain vulnerable to attacks at several different levels.

This chapter identified the vulnerabilities of the operational paradigms currently used by Wireless Sensor Networks and presented security mechanisms to mitigate and lessen those vulnerabilities. An all encompassing generic security model, operating across all operational paradigms, was proffered and suggestions for moving the current state of the art WSN security to that model was suggested.

References

[1] S. Avancha, J. Undercoffer, A. Joshi, and J. Pinkston. A Clustering Approach to Secure Sensor Networks. Technical report, University of Maryland Baltimore County, 2003.

[2] S. Avancha, J. Undercoffer, A. Joshi, and J. Pinkston. Secure Sensor Networks for Perimeter Protection. *accepted for publication in Special Issue Computer Networks on Wireless Sensor Networks*, 2003.

[3] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An On-Demand Secure Routing Protocol Resilent to Byzantine Failures. In *ACM Workshop on Wireless Security (WiSe)*, September 2002.

[4] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti. Secure Pebblenets. In *Proc. of MobiHOC '01*, pages 156–163, October 2001.

[5] S. Bhargava and D. P. Agrawal. Security Enhancements in AODV proocol for Wireless Ad hoc Networks. In *Vehicular Technology Conference*, May 2001.

[6] Stuart Boyer. *SACDA: Supervisory Control and Data Acquisition*. ISA, Triangle Park, NC, January 1999.

[7] D. W. Carman, P. S. Kruus, and B. J. Matt. Constraints and Approaches for Distributed Sensor Network Security (Final). Technical Report 00-010, NAI Labs, 2000.

[8] S. Carter and A. Yasinsac. Secure Position Aided Ad hoc Routing Protocol. In *Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02)*, November 2002.

[9] J. R. Douceur. The Sybil Attack. In *Proc. IPTPS '02*, March 2002.

[10] L. Eschenauer and V. Gligor. A Key Management Scheme for Distributed Sensor Networks. In *Proc. of ACM CCS2002*, November 2002.

[11] X. Fu, B. Graham, R. Bettati, and W. Zhao. On Countermeasures to Traffic Analysis Attack. In *Fourth IEEE SMC Information Assurance Workshop*, 2003.

[12] Y. Guan, X. Fu, D. Xuan, P. U. Shenoy, R. Bettati, and W. Zhao. Netcamo: Camouflaging network traffic for QOS-guaranteed critical applications. *IEEE Trans. on Systems, Man, and Cybernetics Part a: Systems and Humans, Special Issue on Information Assurance*, pages 253–265, July 2001.

[13] R. Housley and W. Arbaugh. Security problems in 802.11-based networks. *Communications of the ACM*, 46(5):31–34, 2003.

[14] Spencer S. Hsu. Sensors may track terror's fallout. The Washington Post. Page A01, June 2, 2003.

[15] L. Hu and D. Evans. Secure aggregation for wireless networks. In *Workshop on Security and Assurance in Ad hoc Networks*, pages 384 – 391, January 2003.

[16] Y. Hu, D. B. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002)*, pages 3–13, June 2002.

[17] Y. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad hoc Networks. In *MobiCom 2002*, September 2002.

[18] Y. Hu, A. Perrig, and D. B. Johnson. Wormhole Detection in Wireless ad hoc Networks. Technical Report TR01-384, Department of Computer Science, Rice University, June 2002.

[19] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proc. of MobiCom '00*, August 2000.

[20] K. Kalpakis, K. Dasgupta, and P. Namjoshi. Efficient Algorithms for Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks. Technical Report TR-CS-02-13, University of Maryland Baltimore County, 2002.

[21] C. Karlof and D. Wagner. Secure Routing in Sensor Networks: Attacks and Countermeasures. In *Proc. of First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[22] B. Krishnamachari, D. Estrin, and S. B. Wicker. Modeling Data-Centric Routing in Wireless Sensor Networks. Technical Report CENG 02-14, Dept. of Computer Engineering, USC, 2002.

[23] Y. W. Law, S. Dulman, S. Etalle, and P. Havinga. Assessing Security-Critical Energy-Efficient Sensor Networks. In *Proc. of 18th IFIP International Information Security Conference*, May 2003.

[24] Robert Lewis. *Modelling Distributed Control Systems Using IEC 61499*. IEE, UK, February 2001.

[25] S. Lindsey, C. S. Raghavendra, and K. M. Sivalingam. Data Gathering Algorithms in Sensor Networks Using Energy Metrics. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):924–935, September 2002.

[26] S. R. Madden and M. J. Franklin. Fjording the Stream: An Architecture for Queries over Streaming Sensor Data. In *Proc. of 18th International Conference on Data Engineering*, February 2002.

[27] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks. In *Proc. of Fifth Symposium on Operating Systems Design and Implementation (USENIX - OSDI '02)*, December 2002.

[28] National Institute of Standards and Technology. *FIPS 46-2; Data Encryption Standard*, December 1993.

[29] P. Papadimitratos and Z. J. Haas. Secure Routing for Mobile Ad hoc Networks. In *Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, January 2002.

[30] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security Protocols for Sensor Networks. *Wireless Networks Journal (WINET)*, 8(5):521–534, September 2002.

[31] J. Raymond. Traffic Analysis: Protocols, attacks, design issues and open problems. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Proceedings of International Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *LNCS*, pages 10–29, 2001.

[32] S. Slijepcevic, M. Potkonjak, V. Tsiatsis, S. Zimbeck, and M. B. Srivastava. On Communication Security in Wireless Ad-Hoc Sensor Networks. In *Proc. of WETICE*, pages 139–144, 2002.

[33] Tony Wilson. Cybercrime's New Foe. The Gold Coast Bulletin (Australia). Page 14, October 25, 2002.

[34] Y. Yao and J. E. Gehrke. The Cougar Approach to In-Network Query Processing in Sensor Networks. *SIGMOD Record*, 31(3):9–18, September 2002.