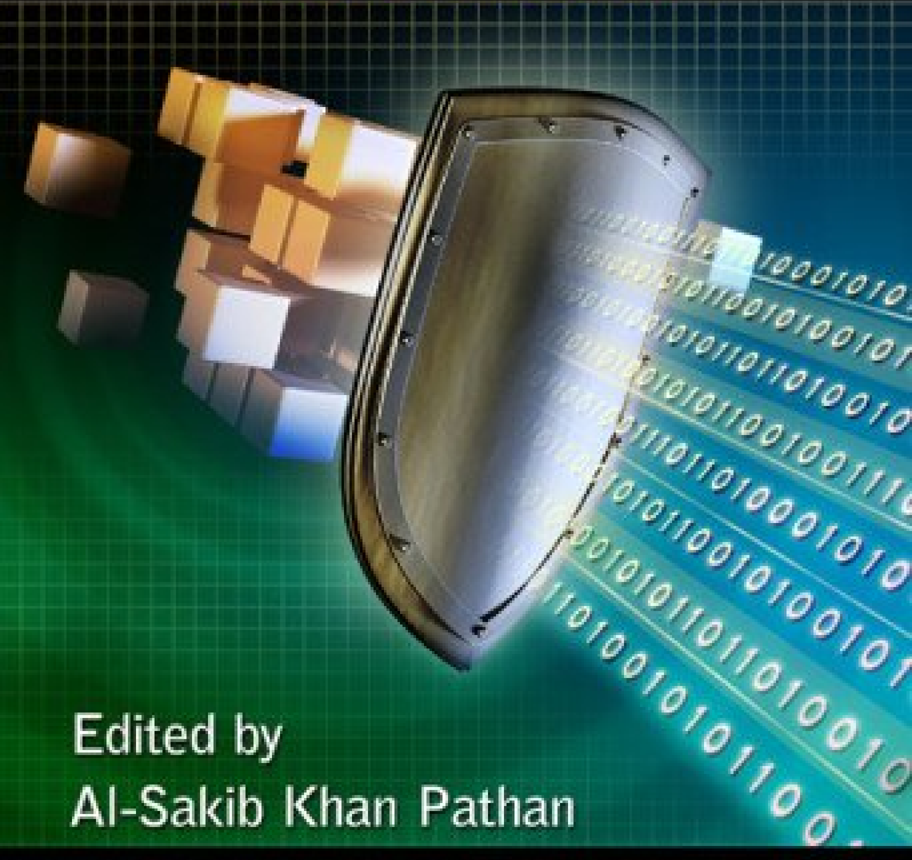
The State of the Art in Intrusion Prevention and Detection



Edited by
Al-Sakib Khan Pathan

CRC Press
Taylor & Francis Group
AN AUERBACH BOOK

The State of the Art in Intrusion Prevention and Detection

Edited by

Al-Sakib Khan Pathan

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20131114

International Standard Book Number-13: 978-1-4822-0351-6 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

The state of the art in intrusion prevention and detection / edited by Al-Sakib Khan Pathan.
pages cm

Contents

PART III Artificial Intelligence Techniques for IDS

PART IV IDS for Wireless Systems

19 An Innovative Approach of Blending Security Features in Energy-Efficient Routing for a Crowded Network of Wireless Sensors

Al-Sakib Khan Pathan and Tarem Ahmed

CONTENTS

19.1 INTRODUCTION

Wireless sensor networks (WSN) are emerging as both an important new tier in the IT (information technology) ecosystem and a rich domain of active research involving hardware and system design, networking, distributed algorithms, programming models, data management, security, and social factors [1,2]. The basic idea of a sensor network is to disperse tiny sensing devices over a specific target area. These devices are capable of sensing certain changes of incidents or parameters and of communicating with other devices. WSNs could be very useful for providing support for some specific purposes, such as target tracking, surveillance, environmental monitoring, etc. Today's sensors can monitor temperature, pressure, humidity, soil makeup, vehicular movement, noise levels, lighting conditions, the presence or absence of certain kinds of objects or substances, mechanical stress levels on attached objects, and other properties. As such types of networks are composed of resource-constrained tiny sensor nodes, many research works have tried to focus on efficient use of the available resources of the sensors. Energy is, in fact, one of the most critical factors that play a great role to define the duration of an active and operable network. Energy efficiency is often very crucial in these sorts of networks as the power sources of the inexpensive sensors are (in most of the cases) not replaceable after deployment. If any intermediate node between any two communicating nodes runs out of battery power, the link between the end nodes is eventually broken. So any protocol should ensure a competent way of utilizing the energies of the sensors so that a fair connectivity of the network could be ensured throughout its operation time. Energy efficiency is also very necessary to maximize the lifetime of the network.

Security, on the other hand, is another critical issue, especially for ensuring the legitimacy of transmitted readings from the sensors to the base station [3,4]. It is anticipated that, in most application domains, sensor networks constitute an information source that is a mission critical system component and, thus, require commensurate security protection. If an adversary can thwart the work of the network by perturbing the information produced, stopping production, or pilfering information, then the usefulness of sensor networks is drastically curtailed. Thus, it should be made sure that the messages from the sensors in action are authentic and reach the base station without any fabrication or modification. As a strong property of security, authenticity of the messages is often considered as the most crucial.

The task of securing wireless sensor networks is, however, complicated, considering the fact that the sensors are mass-produced anonymous devices with a severely limited energy budget and, initially, with no knowledge of their locations in the deployment environment (in general cases). The architectural aspect of wireless sensor networks could make the employment of a security scheme a little bit easier as the base stations or the centralized entities could be used extensively in this case. Nevertheless, the major challenge is induced by the constraint of resources of the tiny sensors. In many cases, sensors are expected to be deployed arbitrarily in the enemy territory (especially in a military reconnaissance scenario) or over dangerous or hazardous areas. Therefore, even if the base station (or sink) resides in the friendly and safe area, the sensor nodes need to be protected from being compromised. At least, it should be made sure that the reports that reach the base station are authentic and are not corrupted on the way of transmission.

In this chapter, we deal with the challenge of energy efficiency and secure routing in wireless sensor networks in a highly dense deployment scenario. We propose a secure energy-efficient routing protocol (SERP) [23], which aims at minimizing the wasteful energy consumption by energy-efficient structuring of the network and then securing the data transmissions from the sensors to the base station using a one-way hash chain and shared secret keys. SERP selects a minimum number of forwarding nodes in the network. It provides a good level of confidentiality and authenticity of the reports sent from the source sensors to the base station.

The major contributions of this chapter are the following:

1. Energy and distance-based efficient structuring of the network, which helps for maximizing the lifetime of the network.
2. Providing data transmission security in wireless sensor networks. Here, we have mainly focused on data authenticity and confidentiality during their transmissions from the source sensors to the base station. There is also an optional key refreshment mechanism in our scheme, which could be applied based on the application at hand to provide data freshness.
3. Detailed analysis and simulation results of our proposed protocol.
4. Overview of security in WSN along with discussion on the impact of different network structures on the security in WSN.

The rest of the chapter is organized as follows: Section 19.2 presents an overview of the threats and attacks against WSNs, Section 19.3 presents the literature review and motivation of this work, Section 19.4 presents our assumptions and preliminaries, Section 19.5 describes our protocol in detail, simulation results and analysis are presented in Section 19.6, and Section 19.7 discusses the possible inclusion of intrusion detection systems (IDSs) based on the network structure and use of SERP as the routing protocol. Finally, Section 19.8 concludes the chapter delineating the achievements from this work with future research directions.

19.2 WSN SECURITY AND THREATS AND ATTACKS AGAINST WSN AT A GLANCE

There are mainly three angles of looking at the security in wireless sensor networks. These angles could cover all the security requirements and issues that we should consider. Figure 19.1 shows a diagram explaining these aspects.

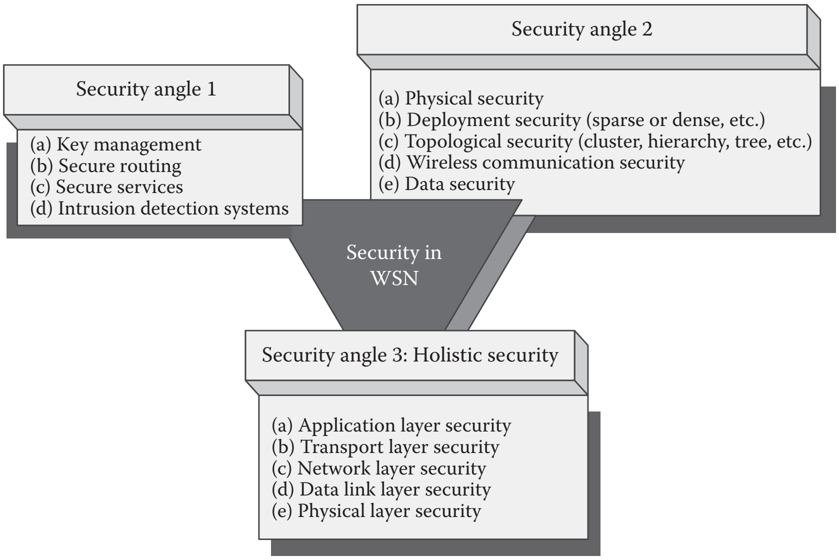

Security angle 1
(a) Key management
(b) Secure routing
(c) Secure services
(d) Intrusion detection systems
Security angle 2
(a) Physical security
(b) Deployment security (sparse or dense, etc.)
(c) Topological security (cluster, hierarchy, tree, etc.)
(d) Wireless communication security
(e) Data security
Security in WSN
Security angle 3: Holistic security
(a) Application layer security
(b) Transport layer security
(c) Network layer security
(d) Data link layer security
(e) Physical layer security

FIGURE 19.1 Three angles of looking at security in wireless sensor networks.

19.2.1 WSN Security Viewing Angle 1

The first angle is based on the mechanism used to deal with security in WSNs. These mechanisms include (a) key management, (b) security routing, (c) secure services, and (d) intrusion detection systems.

19.2.2 WSN Security Viewing Angle 2

The second angle could be based on where the security is employed. This angle includes the following:

(a) Physical security, that is, the physical protection of the sensors in a network, tamper-proof methods, self-destruction method if cracked by attacker, shielding and camouflaging of sensors, etc.

(b) Deployment security, which is dependent on whether the network is sparsely deployed or densely deployed. A densely deployed sensor network may have redundancy in a small area, which could find out alternative ways to protect the traffic flow if attacked by attackers in one way or other. Also, based on the deployment types or the method of deployment of sensors, the security measures may need different types of prior works. If the network is uniformly distributed, the security schemes may be installed uniformly among nodes; again a random deployment may require installing security components in key nodes in the network that cover the entire network.

(c) Topological security: Based on the network structure or network formation, the security could be different. There are mainly three types of network structures: cluster, tree, and hierarchy. In a cluster structure, there is a cluster head in each cluster and some subordinate nodes under the cluster head. In this formation, instead of installing security schemes in each node, cluster heads could be the most suitable entities. This is because of the reason that the cluster heads collect data from the other subordinate nodes and process those before forwarding it toward the base station. If cluster heads with higher computing and energy resources are used in a network, the task becomes easier as they can take the load of processing and forwarding secure packets. If the network formation is tree-based, the nodes have parent-child relationships among themselves from the leaf toward the sink node or vice versa. In such a case, each individual node may include security measures, and along a path in a tree, the packets could be checked before forwarding to the next hop or to the sink node or base station. The third type of network formation is hierarchy, in which there are several hierarchical levels of the nodes in the network. Say, for example, in one level, there are several clusters with cluster heads and subordinate nodes. The cluster heads of this level could be considered as the subordinate nodes in another bigger scale cluster (another level), which might have a higher power cluster head, and it could be repeated for several levels. A well scalable and large WSN with some strategically positioned high power nodes with higher transmission ranges could have such a structure. So such a network formation needs security measures in a different way than the other two types of formations. The thematic diagrams of all these types of network formations are shown later in the chapter when discussing these in relation to our work (see Section 19.7). Other than these categories, there might be hybrid topology in the network, combining different network formation styles, for example, a network with partly a cluster structure and partly a tree-based structure. So it becomes crucial where the security schemes should be installed so that the network security is ensured up to the expected level.

(d) Wireless communication security: Due to the nature of wireless communications, a WSN is always vulnerable. The wireless medium is of an open nature, hence the signaling and reception mechanisms must be secured in the best way possible. An attack such as jamming [25], for example, could disrupt the natural wireless transmissions within the network.

(e) The last category is the data security, which includes the encryption and decryption of data packets, efficient packet authentication techniques, hop-by-hop checking, and so on.

19.2.3 WSN Security Viewing Angle 3

The third angle is the holistic security. This brings forward the concept of layer-wise security in such type of network. Based on the very well known OSI (open systems interconnection) reference model, we could think about ensuring security in each layer. Especially for wireless sensor networks, five layers are relevant: application layer, transport layer, network layer, link layer, and physical layer. Lack of security in any of these levels weakens the overall security of the network. A full working solution in which different mechanisms could work in cooperation is still an open area of research, which would take a huge effort to develop. After knowing all these views and angles of security in a WSN, in the subsequent section, we will explore the major types of threats and attacks against such type of network.

There are several well-known and a few less well-known security attacks that exist in wireless sensor networks. In this section, we discuss these security attacks in brief. Almost all of the attacks described below focus on the limitations of routing protocols in WSNs. However, some unknown attacks that are launched considering other security constraints of the network are presented as well. Table 19.1 introduces a brief summary of well-known and less known (or less studied) security attacks and their characteristics in terms of attack behaviors and techniques.

TABLE 19.1
Security Attacks in WSNs

Well Known		Less Known (or Less Studied)	
Name	**Characteristics**	**Name**	**Characteristics**
DoS attacks in different layers [25–27]	Flooding, jamming, misdirection	Bogus message during reprogramming [28]	Unsecure reprogramming process with bogus messages
Sinkhole/blackhole [29–32]	Shortest path, drop the packets	External stimuli [33]	Use external physical stimuli to create a large number of packets
Selective forwarding [3,34–37]	Selectively drop the packets	Homing [33]	Hamper the normal functioning of cluster heads
The node replication [38,39]	Add extra node to the network with the same cryptographic secrets	Neglect and greed [40]	Deny transmission of legitimate packets and give higher priority to own packets
HELLO flood [41]	Flood with HELLO packets	Unfairness [40]	Unfair resource allocation on MAC protocols
Wormhole [42–45]	Offer less number of hops and less delay, which is fake		
Sybil [36,46–48]	A malicious node pretends to be more than one node		

19.2.4 Denial of Service (DoS) Attacks

We consider any type of intentional activity that can disrupt, subvert, or even destroy the network as a denial of service (DoS) attack.

Basically, DoS attacks can be categorized into three types:

- Consumption of scarce, limited, or non-renewable resources
- Destruction or alteration of configuration information
- Physical destruction or alteration of network resources

These types of DoS attacks are the most significant for WSNs as the sensors in the network suffer from the lack of limited resources. Also, DoSs can be categorized according to the layers of the network architecture. An attacker can use different tools at different layers to stop proper functioning of the entire network or some sensor nodes. Even though it has been said that it is too difficult to know whether any particular DoS situation is caused intentionally or unintentionally, there are some detection methods that exist to thwart each type of DoS attack [72]. In general terms, DoS means any situation that prevents providing proper service that is expected from the network and "DoS attack" means any deliberate activity by an entity (or some) that causes DoS (denial of service) in the network.

Jamming and tampering attacks that exist in the physical layer of WSNs are also considered as kinds of DoS attacks. Jamming is the deliberate interference with radio reception to deny a target's use of a communication channel. Due to their unpredictable nature, WSNs are very vulnerable to "radio channel jamming"–based DoS attack [26]. Tampering is actually any type of physical attack on sensors in the network. They might be physical damage or replacing the sensors, parts of computational or sensitive hardware; one can even extract cryptographic keys to gain unrestricted access to higher communication layers. These types of attacks cannot be defended by some system or base station; only accurate and effective designer of the network can handle it.

19.2.5 Sinkhole or Black Hole Attacks

In this attack, a malicious node acts as a black hole [22] to pull in all the traffic in the network. The attacker listens to the route requests and then replies to the target node informing that it has the shortest path to the base station. A victim node is enticed to select it as a forwarder for its packets. Once the malicious node is able to put itself between the base station and the sensor node, it is able to do whatever it wants (drop packets, change the content, etc.) with the packets that pass through it. This type of attack can be very harmful for sensor nodes that are deployed considerably far from the base station. We have to keep in mind that black hole and sinkhole attacks are basically the same attacks by definition. Some recent works have addressed this attack, and possible IDSs have been proposed in [19,30–32].

19.2.6 Selective Forwarding

Multi-hop networks like WSNs rely on a significant assumption that all nodes in the network will faithfully forward the received messages to the base station (BS). In these attacks, a malicious node acts as a normal node by forwarding only certain messages but selectively drops sensitive packets, which are hard to detect by the system. The specific form of this attack is the sinkhole or black hole attack with which a node might drop all messages it receives. As possible solutions to detect this type of attack, some secure routing algorithms and IDSs using different techniques have been proposed [3,34,35,37].

19.2.7 The Node Replication Attacks

Due to the resource constraints of sensor nodes and often unattended environment of a WSN, an attacker can easily capture the nodes and analyze and replicate them. In this attack, an attacker attempts to add one or more nodes in a network that use the same cryptographic secrets as any other legitimate node in that network. This kind of attack may have severe consequences such as corruption of data by an adversary or even disconnection of some critical parts of the network. Some centralized detection schemes with one point of failure, neighborhood voting protocols with the lack of detecting distributed node replications, and some successful distributed detection techniques have been proposed [38,39].

19.2.8 HELLO Flood Attacks

This attack uses HELLO packets as a tool for convincing the sensors in the network. Many of the routing protocols require broadcasting of HELLO packets to discover the neighbors. An attacker uses this assumption as a weapon to attract the sensor nodes. A node that receives such a packet may assume that it is within normal radio range of the sender node. Hence, an attacker with a large radio range and enough processing power can send HELLO packets to a large number of sensor nodes by flooding the entire network. Thus, the sensor nodes could be persuaded that the adversary is their neighbor. Possible solutions to detect this type of attacks could be the use of bidirectional verification of links before using them, secure multipath routing, and use of multiple base stations [41].

19.2.9 Wormhole Attacks

In this attack, an attacker records the packets at one location in the network and tunnels those to another location. Wormhole attack is another significant and serious threat to WSNs because this is possible even if the attacker has not compromised any node and even if all communications provide authenticity and confidentiality. Attackers offer less number of hops and less delay than other normal routing paths, which leads to attract the sensor nodes to send data through them. While forwarding packets, the attackers can arbitrarily drop sensitive packets. In a recent work, Sharif and Leckie propose three types of wormhole attacks, namely energy depleting wormhole attack (EDWA), indirect wormhole attack (IBA), and targeted energy depleting wormhole attack (TEDWA) [42]. Also, IDS using connectivity information to detect the wormhole attacks has been proposed [44]. Other work proposes a wormhole detection technique using directional antennas, which is, in most of the cases, infeasible for sensor networks due to their limited resources.

19.2.10 Sybil Attacks

In some applications, the sensor might need to work collaboratively to accomplish a certain task; hence, management policy of the network can use distribution of subtasks or redundancy of information. In this case, a malicious node can pretend to be more than one node at the same time using the identities of other legitimate nodes. An attacker tries to degrade the integrity of data, level of security, routing mechanism, data aggregation, and even misbehavior detection techniques. As possible countermeasures, we can use a logically centralized authority (base station or cluster head) in the network. Some recent IDSs could be found in [36,47–49]. Newsome et al. [46] proposed a taxonomy of Sybil attacks in WSNs based on three orthogonal dimensions.

19.2.11 Other Security Attacks in WSNs

There are a few less known (or commonly unknown or less studied) security threats that exist in WSNs. These attacks mostly concentrate on service availability (i.e., DoS) of the networks in different layers. We briefly describe them in the following paragraphs.

Bogus message during reprogramming: This attack could be launched in the application layer if a WSN application allows reprogramming of the network. Reprogramming of the network may be needed for scope selection, encoding-decoding, completion validation, code acquisition, or for network management purposes [28]. If the reprogramming process is not secure enough, the attackers can effectively cut off a portion of the network by using bogus messages.

External stimuli: A possible attack against WSNs in the application layer could be launched by using some external physical stimuli. The attacker uses the external stimuli to stimulate the nodes with a huge number of events to be sent directly to the base station. However, this attack is not effective when packets are sent with predefined regular intervals. The possible solution might be using an IDS that detects attackers in the network if a particular region creates a large number of packets within a short period of time [33].

Homing: Depending on WSN application, some nodes (e.g., cluster heads) are given special responsibilities, such as managing keys, maintaining a local group, etc. The adversaries try to handle and eavesdrop on the activities of those leader nodes. In this attack, the attackers hamper the normal functioning of leader nodes within a WSN application [33].

Neglect and greed: If a sensor node drops packets or denies transmitting legitimate packets or if a node is very greedy to give undue priority to its own messages, then it could be considered as a neglecting node. The protocols that are based on dynamic source routing (DSR) are the most vulnerable to this type of attack [40].

Unfairness: This attack is a weaker form of DoS attack in the link layer. This attack could degrade service for real-time MAC protocols by using unfair resource allocations. In fact, providing fairness in WSNs is often viewed as a separate research issue [40].

So far, we have discussed various types of security threats in WSNs. These attacks can be tackled by using some successful and efficient countermeasures that will be discussed later. Most of the research works basically rely on some statistical assumptions and simulation results. At the time of the implementation of those mechanisms in real environments, they might face plenty of difficulties due to the unpredictable nature of wireless sensor networks.

19.3 LITERATURE REVIEW

We have talked about the major threats and attacks to investigate the grounds of our work a bit. It is, in reality, impossible to tackle all the attacks with a single routing protocol or a single mechanism of any kind (unless different parts of the mechanism work in different layers to cover all the security needs, or different mechanisms work in collaboration to secure the entire network). However, what we can do is the security measures could be blended within the routing mechanism as a first line of defense. Then, on top of that, other security mechanisms could work to deal with specific network-related problems and issues. Hence, the intent of this chapter is to introduce to the readers such a scheme that could give some innovative idea of blending security measures within a routing strategy. There are a few prior works that motivate us to devise our mechanism. Although none of them is directly related to our proposed solution, the underlying principles are sometimes similar to some of them.

Çam et al. [5] propose an energy-efficient security protocol for wireless sensor networks by using symmetric key cryptography and their NOVSF (non-blocking orthogonal variable spreading factor) code-hopping technique. They consider a hierarchical architecture of the network in which data are routed from sensor nodes to the base station through cluster heads. The basic idea of their protocol is to implement two algorithms in the sensor nodes and in the base station, which the sensor nodes and the base station would follow at the time of data transmission and reception. To ensure a better level of security, they introduced the NOVSF technique, which basically scrambles the data blocks using a multiplexer in the system while transmitting data from the sensor nodes. Their scheme is secure and energy efficient, considering the fact that it increases the level of security during data transmission using the NOVSF technique without utilizing any additional power. However, this scrambling technique increases the complexity of tasks for the base station as it has to aggregate and reorder the incoming data blocks correctly. To address the issue of energy-efficient data aggregation with secure data transmission, an ESPDA (energy-efficient secure pattern-based data aggregation) protocol [6] is proposed. In contrast to the conventional data aggregation protocols, ESPDA avoids the transmission of redundant data from the sensor nodes to the cluster head. To make the data transmission and aggregation more secure, a cluster head is not required to decrypt or encrypt the data received from the sensor nodes. On the whole, though, it [5] is an energy-efficient secure protocol; it increases the processing burden of the base station and to support the associated ESPDA scheme, it requires more energy, which literally ruins the gains of the original scheme.

Ye et al. [7] propose a statistical en-route filtering (SEF) scheme to detect and drop false reports during the forwarding process. In their scheme, a report is forwarded only if it contains the message authentication codes (MACs) generated by multiple nodes by using keys from different partitions in a global key pool. According to their findings, SEF can drop up to 70% of bogus reports injected by a compromised node within five hops and reduce energy consumption by 65% or more in many cases.

Zhu et al. [8] propose the interleaved, hop-by-hop authentication scheme that detects false reports through interleaved authentication. Their scheme guarantees that the base station can detect a false report when no more than t nodes are compromised, where t is a security threshold. In addition, their scheme guarantees that t colluding compromised sensors can deceive at most B noncompromised nodes to forward false data they inject, where B is $O(t^2)$ in the worst case. They also propose a variant of this scheme, which guarantees $B = 0$ and which works for a small t.

Motivated by [8], Lee and Cho [9] propose an enhanced interleaved authentication scheme called the key inheritance-based filtering that prevents forwarding of false reports. In their scheme, the keys of each node used in the message authentication consist of the node's own key and the keys inherited from its upstream nodes. Every authenticated report contains the combination of the message authentication codes generated by using the keys of the consecutive nodes in a path from the base station to a terminal node. Other than these works, [10–12] focus only on energy efficiency in a wireless sensor network and the works like [3,4,13] deal with the security measures for routing in WSN.

After analyzing all these works, we design our protocol in which we create a tree structure in the network, based on the energy levels and distances (from the base station) of the sensor nodes. Along with the energy-efficient structuring of the network, we initialize an efficient security scheme down the paths of the tree to ensure secure data transmission in the network. Security is in fact a vast area of research, but our focus of this work is to address secure data transmission from the source sensors to the base station along with energy-efficient structuring and operation of the network. We develop our protocol in a way in which false injection of data cannot deceive the base station or, more specifically, cannot reach the base station. We emphasize the authenticity of sensor readings so that, before transmitting each packet, the forwarding nodes can detect the irregularities with a minimum effort and thus drop unnecessary or flawed packets. By stopping the false packets traveling a long distance along the created paths in the network, our mechanism helps for greater energy efficiency as the intermediate nodes are thus saved from extra transmissions. For employing the entire protocol, we develop it in a way that before starting its operation for secure data transmission, the network is formed in an energy-efficient way. Periodic restructuring of the network is proposed to keep a balance among the nodes to dissipate energies in nearly equal proportion. Our goal here is to achieve maximum lifetime of the network with secure data transmission from any source sensor to the base station.

19.4 ASSUMPTIONS AND PRELIMINARIES

19.4.1 Sensor Deployment and Network Model

We consider a wireless sensor network with densely deployed sensing devices. The deployment could be made by aerial or vehicular scattering or by physical installation. We assume that, initially, all the nodes and the base station in the network have the same transmission range (say r). Like μTESLA [24], our protocol requires that the base station and nodes are loosely time synchronized, and each node knows an upper bound on the maximum synchronization error. The base station has enough energy to support the network's operations for its full lifetime. The sensors deployed in the network have the computational, memory, communication, and power resources such as the current generation of sensor nodes (e.g., MICA2 motes [14]). Once the sensors are deployed over the target area, they remain relatively static in their respective positions. That means the nodes do not move with respect to their neighbor. The transmissions of each node are isotropic (i.e., in all directions) so that each message sent is a local broadcast within the transmission range of the node. The link between any pair of nodes in the network is bidirectional, that is, if a node n_i gets a node n_j within its transmission range (i.e., one hop), n_j also gets n_i as its one-hop neighbor.

An accurate model for the energy consumption per bit at the physical layer is given by

$$E = E_{elec}^{trans} + \beta d^{\alpha} + E_{elec}^{recv} \tag{19.1}$$

where E_{elec}^{trans} is the distance-independent amount of energy consumed by the transmitter electronics (PLLs, VCOs, bias currents) and digital processing, E_{elec}^{recv} is the energy utilized by receiver electronics, and βd^{α} accounts for the radiated power necessary to transmit over a distance d between source and destination. As in [15,16] we assume that

$$E_{elec}^{trans} = E_{elec}^{recv} = E_{elec} \tag{19.2}$$

So overall energy consumption between source and destination within one hop can be calculated using

$$E = 2.E_{elec} + \beta d^{\alpha} \tag{19.3}$$

Broadly speaking, hierarchical routing protocols use control packets for topology construction phase. For a particular node i, control packet transmission cost can be calculated by

$$C_i^{ctrl}(r) = \left[L_{ctrl} \times \beta r^{\alpha} + (nbr_i(r)+1) \times L_{ctrl} \times L_E\right]\frac{1}{T} \tag{19.4}$$

where, α is the path loss exponent ($2 < \alpha < 5$), β is a constant [joule/bit.m^2], r is the transmission range, L_{ctrl} is the length of control packet in bits, nbr_i is the average number of neighbors of node i for range r, L_E is the energy needed by the transceiver circuitry to transmit or receive a packet, and T is the time period between two consecutive restructurings of the network.

For a particular path p, data communication cost from source i to the base station can be represented as

$$C_i^{data}(p) = \left[\sum_{i=1,j=2}^{N} (nfrd_p(d_i)+1) \times L_{data} \times \beta d_{i,j}^{\alpha} + (nbr_p(d_i)+1) \times L_{data}\right] \times L_E \tag{19.5}$$

Here, N is the total number of nodes in the network, $i,j \in 0, 1, 2, ..., N$ is the node index, p is the path associated for data transmission from source i to sink, d_i is the transmission range set by node i, $d_{i,j}$ is the distance between the node i and j, $nfrd_p$ (d_i) indicates the number of forwarding nodes for a path p and range d, nbr_p (d_i) indicates the number of neighboring nodes for a path p and range d, L_{data} is the length of data packets in bits, and, finally, α and β are same as the previous equation. Total communication cost for sending a data packet from source i is

$$C_i^{total}(p) = \sum_{i=1}^{N}\left[C_i^{ctrl}(r)\right] + C_i^{data}(p) \tag{19.6}$$

The observations from the above equation are the following:

- Wasteful (due to idle listening, overhearing, etc.) energy consumption increases as the number of redundant forwarder increases.
- Wasteful energy consumption increases as the number of idle nodes increases.
- Energy consumption increases exponentially as the distance between nodes increases.
- Frequency of control packet transmission is proportional to the energy consumption.

To reduce energy consumption, the following things could be done:

- Reducing the number of forwarding nodes (not hampering the level of connectivity and the reliability of the network)
- Putting a certain portion of the nodes in sleep mode to reduce idle mode energy consumption
- Employing adaptive transmission range according to the distance from the forwarder node to save energy
- Fixing the network restructuring frequency to ensure balanced energy consumption

19.4.2 Basic Terms and Definitions

We consider three states of the nodes in our protocol during its operation:

Non-forwarding – Nodes keep their radio transceivers "off" but continue to sense the events in their sensing ranges using sensing circuitry. Sensing of any event turns on the radio of a non-forwarding node.

Forwarding – Both the transceiver and sensing circuits remain "on" in this state.

Active – During the tree construction and OHC initialization phase (later described in Section 19.4.1), all nodes remain in the active state. In the active state, both the sensing and radio circuitries of the sensors remain "on." Basically, there is no major difference between forwarding and the active state. We term these two states to differentiate the two phases in our protocol (explained later).

Active State Time. Let v be a node and $N_1(v)$ be the number of one-hop neighbors of v for a particular transmission range r (r is same for all nodes in the network, including the sink). Let T_{rtt} be the round trip time for data propagation between the longest distant pair within one-hop neighbors. Then, the active state time for node v is given by the equation

$$T_{active} = T_{rtt} \times N_1(v)$$

In our protocol, within the time T_{active}, a node could be able to determine whether it should participate in the tree as a forwarding node or not.

One-way Hash Chain (OHC). To ensure security for data transmissions from the sensors to the base station, we use pre-stored shared secret keys and a one-way hash chain. A one-way hash chain [17] is a sequence of numbers generated by one-way function F that has the property that for a given x, it is easy to compute $y = F(x)$. However, given F and y, it is computationally infeasible to determine x, such that $x = F^{-1}(y)$. A one-way hash chain (OHC) is a sequence of numbers K_n, K_{n-1}, …, K_0, such that, $\forall$i : $0 \leq i < n$, $K_i = F(K_{i+1})$. To generate an OHC, first a random number K_r is selected as the seed, and then F is applied successively on K_r to generate other numbers in the sequence. In the next section, we describe in detail how the shared secret keys are used with OHC in our protocol to provide data transmission security.

It should be noted here that in this chapter we have used the terms *base station* and *sink* interchangeably.

19.4.3 Security Assumptions and Threat Model

The base station could not be compromised in any way. We assume that no node could be compromised by any adversary while creating the tree structure in the network (i.e., the first phase of our scheme). This particular assumption is necessary to protect the network from being wrongly structured or to prevent the inclusion of any rouge entity in the network. In this case, we are mainly assuming that compromising a node with physical capture is not possible. Also some other attacks, such as jamming, could hamper proper relaying of the control messages. We assume that, at least in the tree structuring process, any physical capture or jamming attack is not done by any adversary. In fact, such types of initial attacks (for example, Hello Flood attack [3]) could be another topic for research. In this chapter, our focus is to secure the data transmissions from the source sensors to the base station, and addressing jamming or physical capture are beyond the scope of this work. Initially, each node is equally trusted by the base station.

Each node in the network has a unique shared secret key with the base station. These keys are pre-stored into the sensors' memories so that, after deployment, the sensors could use the keys to encrypt data while sending it to the base station. The base station keeps an index of the IDs of the sensors and the corresponding shared secret keys. Due to the use of wireless communications, the nodes in the network are vulnerable to various kinds of attacks. We assume that an adversary could try to eavesdrop on all traffic, inject false packets, and replay older packets. If, in any case, a node is compromised, it could be a full compromise in which all the information stored in that particular sensor are exposed to the adversary or could be a partial compromise, that is, partial information is exposed.

19.5 SECURE ENERGY-EFFICIENT ROUTING PROTOCOL (SERP)

19.5.1 Tree Construction and OHC Initialization Phase

We consider distances and residual energies of the nodes to construct a sink rooted tree (SRT) in the network. At the time of the tree construction, all nodes keep their radio transceivers "on" to verify whether it should remain active as a forwarding node or not. A timer parameter is defined to ensure each node's active participation in this process for a specific period of time. Each node is prioritized for transmission according to its residual energy and distance from the sink.

Now, according to our assumption, all the sensors and the base station have shared secret keys that are pre-stored before deployment of the network. So when the sensors are deployed in the target area randomly, each sensor contains a shared secret key with the base station, which could be used to provide confidentiality of the reports. However, to provide authenticity of the transmitted data, all the intermediate nodes between any source node and the base station must be initialized with the basic one-way hash chain number. Let us suppose the initial OHC number is $I_{OHC} = HS_0$.

To initiate the first phase of network structuring and OHC number initialization, the base station B generates a control packet containing HS_0, a MAC (message authentication code) for the control packet using the key K_i along with some other parameters. Here, K_i is the number in the key chain corresponding to time slot t_i. The format of the control packet is

$$bcm\text{: } B|sid|ren|dist|fid|HS_0|\mathrm{MAC}_{K\mathrm{i}}(B|sid|ren|dist|fid|HS_0)$$

where, *sid* indicates the sender's ID, *ren* is the remaining energy of the sender, *dist* is the calculated cumulative distance to reach the sink using forwarding node(s), and *fid* is the ID of the upstream node (i.e., immediate parent or immediate forwarding node) selected by the current node for forwarding data toward the sink. The sink node initiates *bcm* with sender ID *B*, and the values of *sid*, *ren*, *dist*, and *fid* as –1 as, according to our assumption, the base station has unlimited energy compared to the energies of the sensors in the network, and in this case, no forwarding node is needed to reach itself.

When the BS transmits *bcm*, at first, its one-hop neighbors get the message. Receiving the message, each node in the one-hop neighborhood of the base station first calculates its distance (i.e., *dist*) from the base station based on the received signal strength, stores the value of HS_0 and sets *B* as its forwarder node (the ultimate destination is the base station). Now, each of these nodes transmits the message again within its own one-hop neighborhood (i.e., local broadcast). In this case, the *sid* is set to its own ID, *ren* is its own residual energy, and the MAC part is kept the same as the base station message, *bcm*. To ensure prioritization of the transmission of control messages, each node waits for a threshold time before each further transmission. Waiting time of a node before further transmission is defined by

$$T_{wait} = \{D_s/E_r\} \times R \tag{19.7}$$

where D_s is the cumulative distance between the sink and the node, E_r is the residual energy of the node, and R is a constant that is needed to normalize the value of T_{wait}. As with the course of time, the sensors lose their energy levels, and the value of the ratio of distance and residual energy increases; we need to normalize this value. In our case, R is the ratio of the node's initial energy and transmission range.

Each node receiving the control messages from one or more upstream neighboring nodes first calculates the distance of each sender based on the received signal strength, then calculates the cumulative distances up to the sink via different possible forwarders (i.e., the upstream senders), stores the ID and residual energy information of each sender, and stores HS_0 from the message sent by the first sender. To choose its forwarder node, it compares the values of the distance and energy ratios (D_s/E_r) of the neighboring upstream nodes and chooses the node with the least value of the ratio as its forwarder node. It then senses the channel, and if the channel is idle, it waits for T_{wait} time and then retransmits the message containing its own status information and with its chosen forwarder node ID as its *fid*. As the selected upstream node could also get the message (as the link between any two nodes is bidirectional), it sets itself as a forwarding node for this transmitting node. This process continues, and eventually a tree structure is created in the network in which each node has a forwarder node on the way to reach the base station and possibly one or more downstream nodes that can send data to it destined to the base station. Here, the value of T_{wait} depends mainly on the values of E_r and D_s. In fact, these values are used to set the priority of the nodes to be selected as forwarding nodes.

To authenticate HS_0, B releases the key K_i in time slot t_{i+d}. Here, d is the delay parameter for the time slot, which could be set depending on the application at hand. It indicates after how many time slots the key for time slot i should be released. On receiving this key, a node can verify the integrity and source authentication of HS_0. Thus, along each path, the initial OHC number is initialized. It is to be noted that *bcm* won't bring any attack against the network even if the nodes on the other side of the network don't receive K_i at t_{i+d}. Because the messages that are MACed by K_i are supposed to be sent out at time slot t, an adversary cannot launch any attacks with K_i when it gets K_i at t_{i+d}. Within the time T_{active}, a node that does not get any message from any of its neighbors that it should be a forwarding node, sets itself as a non-forwarding node. Figure 19.2 shows the sample input and output networks.

19.5.2 Network Operation and Secure Data Transmission Phase

We construct the sink rooted tree (SRT) based on the energy levels and distances of the nodes. After the tree is constructed within the network, all nodes are either in forwarding or non-forwarding

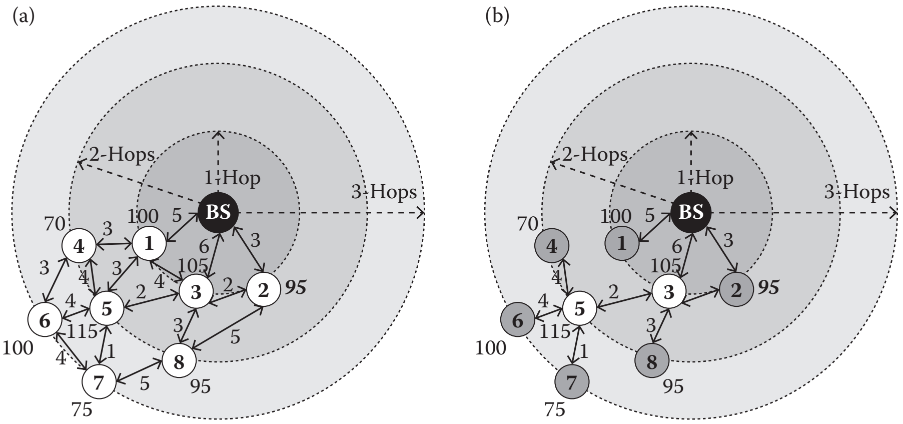


FIGURE 19.2 A portion of an example network (a) before execution of the first phase. All the white nodes are in active status (b) after execution of the first phase. The gray nodes are in non-forwarding status while the other two nodes are in forwarding status. We have shown the *N*-hop (*N* = 1, 2, 3, …) neighbors of the sink on the circumference of the same circle regardless of their actual calculated distances from the sink.

states. Nodes with the non-forwarding state turn off their radio transceivers while keeping the sensing circuitry "on." On the other hand, forwarding nodes keep both radio and sensing circuitry "on." All nodes try to sense any change of parameters (such as temperature, pressure, magnetism, etc. based on the duty assigned to the nodes) within their vicinities, and upon detecting any event, the non-forwarding nodes turn their radios on and transmit data toward the base station via their selected forwarding nodes.

To send the data securely to the sink, each source node *ns* maintains a unique one-way hash chain, HS: $< HS_n, HS_{n-1}, \ldots, HS_1, HS_0 >$. When a source node n_s sends a report to the sink using the path created in the sink-rooted tree (for example, a path is $n_s \rightarrow \ldots \rightarrow n_{m-1} \rightarrow n_m \rightarrow B$), it encrypts the packet with its shared secret key with the base station, including its own ID and an OHC sequence number from *HS* in the packet. It attaches HS_1 for the first packet, HS_2 for the second packet, and so on. To validate an OHC number, each intermediate node $n_1, \ldots, n_m$ maintains a verifier I_{n_s} for each source node n_s. Initially, I_{n_s} for a particular source node is set to HS_0. When n_s sends the *i*th packet, it includes HS_i with the packet.

When any intermediate node n_k receives this packet, it verifies whether $I_{n_s} = F(HS_i)$ or not. If so, n_k validates the packet, it forwards it to the next intermediate node, and sets I_{n_s} to HS_i. In general, n_k can choose to apply the verification test iteratively up to a fixed number *w* times, checking at each step whether, $I_{n_s} = F(F \ldots (F(HS_i)))$. If the packet is not validated after the verification process has been performed *w* times, n_k simply drops the packet. By performing the verification process *w* times, up to a sequence of *w* packet losses can be tolerated, and the value of *w* depends on the average packet loss rate of the network. An intermediate node need not decrypt the packet; rather it can check the authenticity of the packet before forwarding to its immediate forwarder. Figure 19.3 illustrates these.

In Figure 19.3a, the source node *ns* sends the first packet to the base station with the OHC value HS_1. The content of the packet is encrypted with the secret key that it shares with the base station. Getting the packet, the base station performs the authenticity check by verifying the hash chain number and gets the report by decrypting it with the shared key for that particular source node. Figure 19.3b shows a scenario in which the packet P_2 could not reach the base station for some reason. In spite of that, the OHC verification is not hampered as for the next packet the third intermediate node performs the hash verification twice (Figure 19.3c). Here, at the very first attempt ,it cannot get the value of HS_1 in the verification process, but in the second iteration, it verifies it as a valid packet from the source n_s. In fact, in this case, the intermediate node can

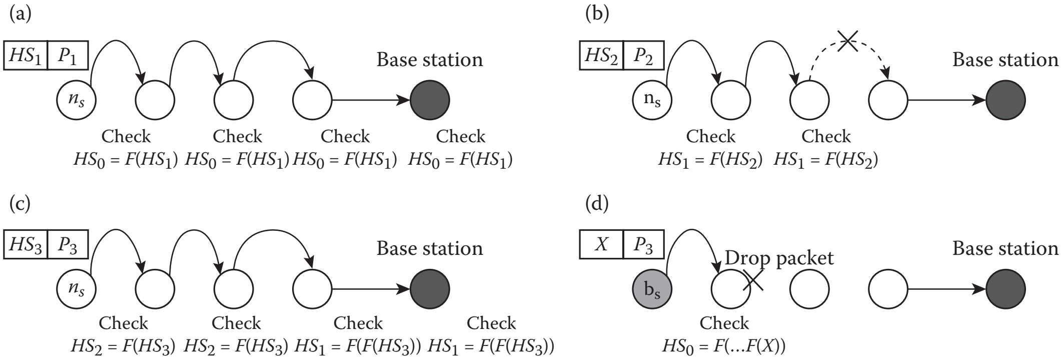


FIGURE 19.3 (a) Authenticated packet delivery to the base station using the OHC numbers, (b) an example scenario in which the packet could not reach the base station, (c) but it cannot affect the OHC verification technique, (d) a bogus packet with a false *HS* value is dropped by an intermediate node.

perform the hash number verification *w* times, and *w* is an application-dependent parameter. In Figure 19.3d an adversary tries to send a bogus packet with a false hash chain number, and it is detected in the next upstream node. Eventually, such a bogus packet fails to pass the authentication check and is dropped in the very next hop. This feature saves energy of the network as such falsely injected packets cannot travel through the network for more than one hop. After the tree construction, at the time of data transmission, each node could dynamically set its transmission range according to the distance of the parent or immediate forwarding node. If the distance of the forwarding node is less than the initially used transmission range for tree construction, the node decreases the range by decreasing the transmission energy. This feature gives the flexibility in our protocol to dynamically set the transmission ranges, and thus it helps for conserving network-wide energy.

The first phase is executed after every *T* time, and *T* is an application-dependent parameter. *T* depends on the event generation rate as well as on the load of the network. Each node participating in tree construction should have at least a certain level of energy.

19.5.3 Optional Key Refreshment

To provide data freshness and to increase the level of security, our scheme has an optional key refreshment mechanism. In this case, the base station periodically broadcasts a new session key to the sensors in the network. The format for this message is

$$B|K_s|\,\mathrm{MAC}_{Kj}(B|K_s))$$

where K_j is the number in the key chain number corresponding to time slot t_j. To authenticate K_s, like the OHC initialization phase, B releases the key K_j in time slot t_{j+d}. On receiving this key, the nodes can verify the integrity and source authentication of K_s. Then each node gets the new key by performing an X-OR (exclusive OR) operation with its old shared key. This method could also be utilized for refreshing the keys of a specific number of nodes. In that case, the base station could simply send the K_s to the specific node by encrypting it with its previous shared secret key. Upon receiving the new key, the node can perform the X-OR operation and could use the newly derived key for subsequent data transmissions.

Changing encryption keys from time to time has an advantage as it guarantees data freshness in the network. Moreover, it helps to maintain confidentiality of the transmitted data by preventing the use of the same secret key at all the times.

19.5.4 Repairing a Broken Path and OHC Re-Initialization

If, in any case, any node between the source node and the base station fails, it could make one or more paths useless. Eventually, in such a case, all the downstream nodes along that particular path get disconnected from the base station. To repair such a broken path, we use the stored upstream knowledge of the sensors. We know that, in the first phase, each downstream node stores the IDs of the one-hop upstream senders of the control message. So this knowledge could be used for repairing the path quickly.

Let us illustrate it with an example. Say, in Figure 19.2b, node 5 is somehow damaged or failed to continue (Figure 19.4a). So the nodes 4, 6, and 7 get disconnected from the base station. This failure could first be detected by the one-hop neighbors of node 5 in the tree, i.e., nodes 4, 6, 7, and node 3. In the first phase, as node 4 got a message from node 1, which tried to become its forwarder, node 4 could use that knowledge to repair the path. So node 4 first does a local broadcast of an error message that it has lost its previous forwarder and sets node 1 as its forwarder. Accordingly, node 1 gets a forwarding status. If there were more senders who had sent control messages to node 4 at the time of tree construction, node 4 would have chosen the node with the least distance and energy ratio as recorded earlier. We know that in the first phase, each node stores the information about its neighbors who try to become its forwarder. If node 4 is required to send any packet as a source node, it could simply send it using the OHC number in the sequence, HS_{k+1}, which is next to its last-used OHC number, HS_k. For node 1, node 4 is a new source, so it could save its HS value in I_4. The subsequent transmissions from node 4 are verified by node 1 based on this initial knowledge. There are other two stranded nodes in our example, node 6 and node 7. In the similar fashion, these nodes use their stored knowledge. The structure of the new path after broken path recovery is shown in Figure 19.4b.

As we are considering a highly dense deployment scenario, we think that, in most of the cases, a node might initially get two or more upstream senders who would try to be its forwarder. This procedure works fine as long as no more than w packets are lost on the way from any source node (after a path is broken due to a node failure). If, within the time of repairing the path, more than w packets are lost from a particular source, the OHC chain along that path breaks down. In fact, this is the worst case in which all the downstream nodes along the path become invalid to the base station and their sent data are discarded on the way to reach the base station. To overcome this problem, the entire OHC initialization phase (the first phase of our protocol) could be made periodic (after an certain interval, which is an application-dependent parameter). Determining the best possible time interval for re-initialization of the first phase is kept as our future work.

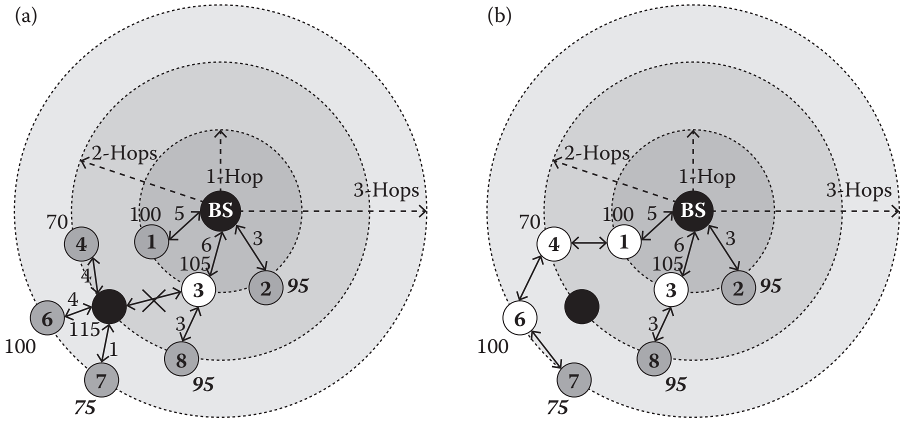


FIGURE 19.4 (a) Node 5 failed; (b) repairing a broken path. White nodes are in forwarding status, and gray nodes are in non-forwarding status.

19.6 SIMULATION RESULTS AND PERFORMANCE ANALYSIS

19.6.1 Simulation

To understand the performance of our proposed protocol, we simulated the network in NS-2 [18] with 50 to 300 nodes uniformly distributed in a 100 m × 100 m square sensor field. The transmission range of each sensor node was set to 25 m. Each node was provided with 2 Joule of initial energy. Transmitter and receiver electronics were set to dissipate 50 nJ/bit.m^2. The data packet length was set to 2 KB. The sink or base station was located at (150, 150) coordinate. We varied the number of sources from 1 to 7, and the data generation interval was randomly chosen. Initially, tree construction time was set to 10 seconds. As our protocol creates a hierarchical structure in the network, we compared our protocol with two other hierarchical energy-aware routing protocols LEACH [10] and EAD [11]. All the simulation parameters are shown in Table 19.2.

After the construction of the sink rooted tree, some of the nodes are selected as the forwarding nodes. The size of the set of forwarding nodes indicates at least how many nodes are needed to stay awake for data transmission. A small set of forwarding nodes is desirable for minimizing the routing overhead. The smaller the size of the set of forwarders, the better the energy efficiency is for the network as more nodes could be in the non-forwarding status. Figure 19.5a shows the percentage of forwarding nodes among the total number of nodes in LEACH, EAD, and our protocol. Now, an interesting feature to note for the Figure 19.5a is that as the number of nodes in the network grows, the percentage of cluster heads decreases slightly for LEACH because more nodes become associated with a single cluster head in the network. For the reason of dense deployment, relatively more nodes are covered by a cluster head. Thus, the percentage of cluster heads (forwarding nodes) becomes slightly lower than the suggested percentage of cluster heads as the number of nodes increases in the network.

Figure 19.5b shows the energy dissipation given a number of source nodes. Less energy dissipation eventually helps for increasing the lifetime of the network. The relative gain of our proposed scheme compared to LEACH and EAD increases with the increase of number of sources. More sources issue more data to be transmitted. In case of LEACH, each transmission requires one hop to reach the cluster head and one hop to reach to the sink. In case of EAD, multiple hops are required to reach to the sink. As wireless transmission power varies depending on distance, for the same packet size, LEACH requires much higher energy for transmission. EAD requires less energy than that of LEACH as it uses multiple hops (hence, less transmission range). As our algorithm uses adaptive transmission range, the amount of energy consumption is much less than LEACH and EAD, considering the same packet size.

TABLE 19.2
Simulation Parameters

Simulation time	1300 s
Simulation area	100 × 100 m^2
Total number of nodes	50 ~ 300
Initial energy	2 J
Transmit/receive electronics (L_E)	50 nJ/bit/m^2
Transmission power	5.85 e – 5 watt
Receive signal threshold	3.152 e – 20 watt
Sleep mode energy	0
Number of sources	1 ~ 7
Offered load	4 ~ 6 pkts. per s (pps)
Transmission range	25 m
Packet size	2048 bytes

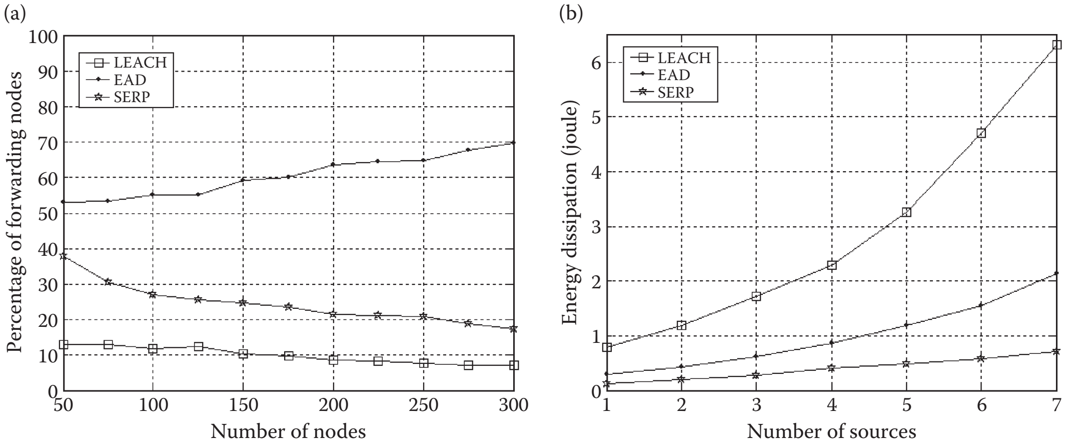


FIGURE 19.5 (a) Percentage of forwarding nodes in total number of nodes in the network, (b) energy dissipation for different number of sources in LEACH, EAD, and SERP.

Figure 19.6a and 19.6b present the number of *alive* nodes versus simulation time with 50 and 100 nodes. Our proposed scheme generates a fewer number of forwarding nodes compared to EAD. As a result, the energy dissipation is much less than that of EAD as there are less nodes participating actively in the network operation phase. Also adaptive transmission range saves more energy for the same packet size. Single hop transmission, the main drawback of LEACH, leads to huge energy consumption for data transmission. Our experimental results show that our algorithm achieves better lifetime compared to LEACH and EAD.

19.6.2 Storage Requirement for One-Way Hash Chain

The method of generating and storing a long OHC in a sensor node is a little difficult. Naive algorithms require either too much memory to store every OHC number or too much time to compute the next OHC number. None of these algorithms are practical on resource-constrained sensor nodes. Recently, some efficient OHC generation algorithms for resource-constrained platforms have been

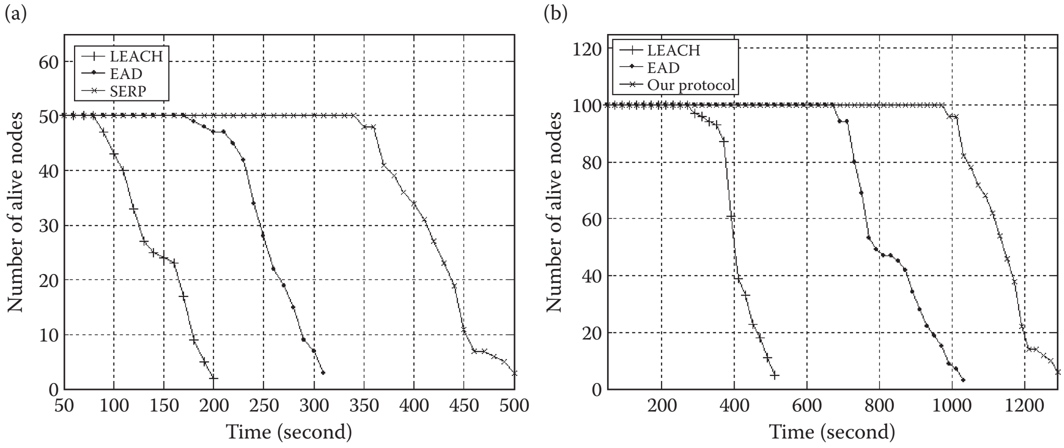


FIGURE 19.6 Number of alive nodes versus time for (a) 50 nodes and (b) 100 nodes.

proposed [19–21]. Among these algorithms, the fractal graph traversal algorithm [19] could perform well on the traditional sensor nodes. This algorithm stores only some of the intermediate numbers, called pebbles, of an OHC, and uses them to compute other numbers. If the size of an OHC is n (there are total n numbers in this OHC), the algorithm performs approximately $\frac{1}{2}\log_2 n$ one-way function operations to compute the next OHC number and requires a little more than $\log_2 n$ units of memory to save pebbles.

The length of an OHC that is needed for a source node is also an important factor. The typical length is between 2^{11} to 2^{22}. If the length of an OHC is 2^{22}, and a node uses one OHC number per second, it will take more than a month to exhaust all numbers from this chain. Figure 19.7a shows the storage requirements for storing pebbles for different lengths of an OHC. This includes a skipjack-based one-way function and OHC generation based on [19]. We see that a node needs about 930 bytes to maintain an OHC of length 2^{22}. This includes a 256-byte lookup table for skipjack, which can be shared with other applications. Other than this, each node has to store only a few IDs and neighbor information of its one-hop neighbors. Overall, the memory requirement for our scheme could be well afforded with today's sensor nodes.

19.6.3 Security Analysis

We analyze the security of our scheme with respect to two design goals: the ability of the base station to detect a false report and the ability of the nodes en route to filter or detect false reports.

Base Station Detection: In our scheme, whenever the base station receives a report from any sensor, it first checks the ID of the sensor, checks the authenticity of the report by verifying the one-way hash chain number for that particular source, looks for the corresponding shared secret key, and decrypts the packet. The base station could not be compromised in any way. So it is in fact the final entity that could confirm the authenticity, confidentiality, and integrity of the transmitted reports. Our security scheme is designed in a way that, any bogus report cannot reach the base station; rather it would be detected and dropped by the intermediate nodes. However, if, somehow, a bogus packet is sent directly to the base station, it would certainly be discarded by it for the failure of the authentication check. If in any application, the optional key refreshment mechanism is employed, once the time slot of releasing the new session key is over, the base station first tries to decrypt the incoming packets from any particular source with the X-ORed new key for that node. In case it produces a garbage result, the base station tries with the previous shared secret key with that node (the previous key could easily be obtained again by X-ORing the most recent session key with the

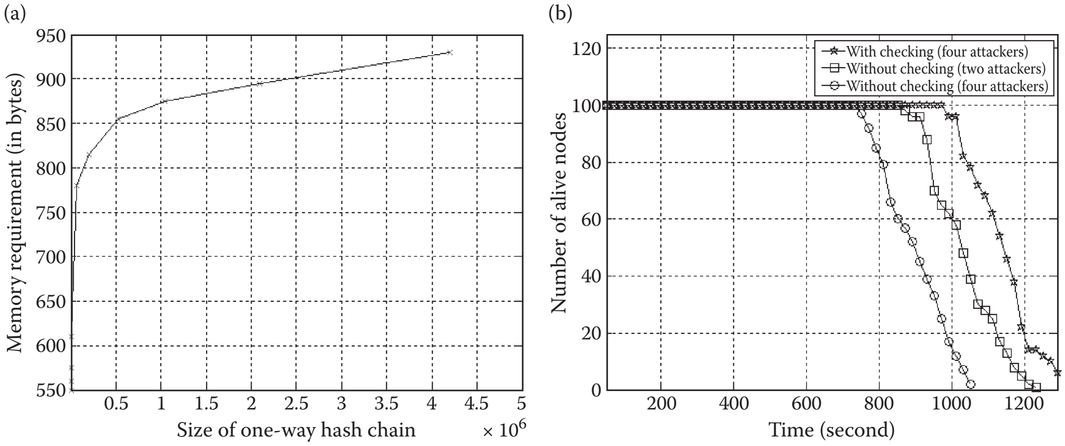


FIGURE 19.7 (a) Memory requirement for OHC generation; (b) Number of alive nodes versus time for two cases: without packet authentication and with packet authentication (2 ~ 4 attackers generating bogus packets).

newly computed key for that node). This case might happen when somehow some node cannot get the new session key released by the base station.

Detection by the Intermediate Nodes: Two types of attempts from the adversaries are considered:

Outsider Attack: In this case, as shown in Figure 19.3d, if an outsider node generates a packet with a fake OHC number, the authentication must be failed in the very next node in the path, and as a result, this packet would never be forwarded even to the node that is only two hops away from it. Simple verification of the OHC number prohibits the forwarding of such bogus packets.

Insider Attack: If a legitimate node along any path is compromised, the attacker could grab the OHC sequence and the shared secret key with the base station. However, it should be noted that, to use the OHC numbers successfully, the adversary should also know the last OHC number used by that particular node to send the packet to the base station. If it gets the last-used OHC number, then it could use this for sending false packets successfully. Otherwise, any arbitrary use of the OHC number from that source might not be forwarded by the next intermediate node because of authentication failure. Now, in case of a node that is fully compromised, that is, if the adversary obtains all the required information, it actually gets the status of a legitimate node in the network. This fully compromised node could be used to generate false reports with valid authentication numbers. To prevent such type of malicious adversary, there are several factors that come into play to detect the abnormal behavior of the node. In our scheme, the base station considers a report legitimate if it is reported by at least δ number of source nodes in the network, where δ is an application-dependent parameter. So the different or modified reports from a single source cannot convince the base station about any event. Also the base station could notice the amount of packets generated by a particular source. These are basically a part of an intrusion detection system (IDS) implemented in the base station. The detailed description of the IDS is beyond the scope of this work.

The worst case scenario occurs if more than δ number of nodes in a particular region in the network are somehow compromised. This sort of collaborative and large-scale attack is handled by the periodic restructuring of the whole network. Finding an optimal value of the time interval for periodic restructuring is kept as our future works.

In Figure 19.7b, we show the number of alive nodes versus simulation time considering the packet authenticity checking method and without checking. We considered two to four attackers in addition to the number of actual source nodes. The graph shows that if the detection method is absent, the nodes lose energy rapidly, which causes a shorter network lifetime. The result is plotted for total of 100 nodes in the network. In this experiment, four to 16 packets per second (pps) were generated by the attackers to drain energy of the nodes. When the packet authentication method is employed, the nodes can detect false packets, and by dropping those other intermediate nodes, are relieved from the burden of forwarding false reports.

As a whole, the efficiency of our protocol is increased with the number of false packets transmitted by the attackers. The more false packets that are tried to be sent by the adversaries, the more gain we have as those packets cannot travel a long distance toward the base station and thus save the network from consuming unnecessary energy by extra forwarding or transmission. This is, in fact, very helpful for the longer lifetime of the network in a heavy flooding attack in which the attackers try to inject a huge number of false packets into the network data flow.

19.7 DISCUSSION ON IMPLEMENTING AN IDS WITH SERP

As we noted earlier, the description of a complete IDS is beyond the scope of this chapter. However, in this section, we briefly discuss the implementation of an IDS alongside SERP, considering different WSN structures. Depending on the network structure used, the location of the employed IDS could be different, which could also affect various parameters in the network. The objective of putting this section is to link up some IDS techniques that could be considered for a network in which SERP is used as the routing protocol. While SERP provides partial protection by providing authentication of packets and minimizing the energy drain, any IDS in particular locations of the network could give the rest of the protection that the network needs for its overall security.

WSN is a highly application-dependent network. Hence, network structures vastly differ depending on the application types. We have discussed the major structures in the introductory section in a more detailed way. However, in this part, for the convenience of the readers, we are recapping the gist of the previously noted information to relate IDS with current discussion:

- *Tree based* – In this structure, the base station plays the role of the main parent node (i.e., root), and sensor nodes take the roles of leaf nodes or intermediate nodes. The one-hop neighbor nodes of the base station can become parent nodes for the second hop neighbor nodes, and this method continues to cover the entire network in this fashion.
- *Cluster based* – In this scenario, the network is divided into clusters with the main base station. Every cluster has its own selected cluster head (CH), which is the medium between cluster members and the base station. In addition, cluster heads are often allowed to communicate among themselves for some specific purposes.
- *Hierarchical* – The network is organized into a tree-like structure with several different types of clusters in it. This structure may have several layers representing parent-child type relationships (at least thematically). Note that this is different than a hybrid model in which a portion of the network is cluster based while some other portion is tree based and some other portion may be of hierarchical structure or combination of all.

Figure 19.8 illustrates these network structures with possible IDS locations at which IDSs can function in a perfect and efficient manner. For instance, in a tree-based structure, it would be perfect if the

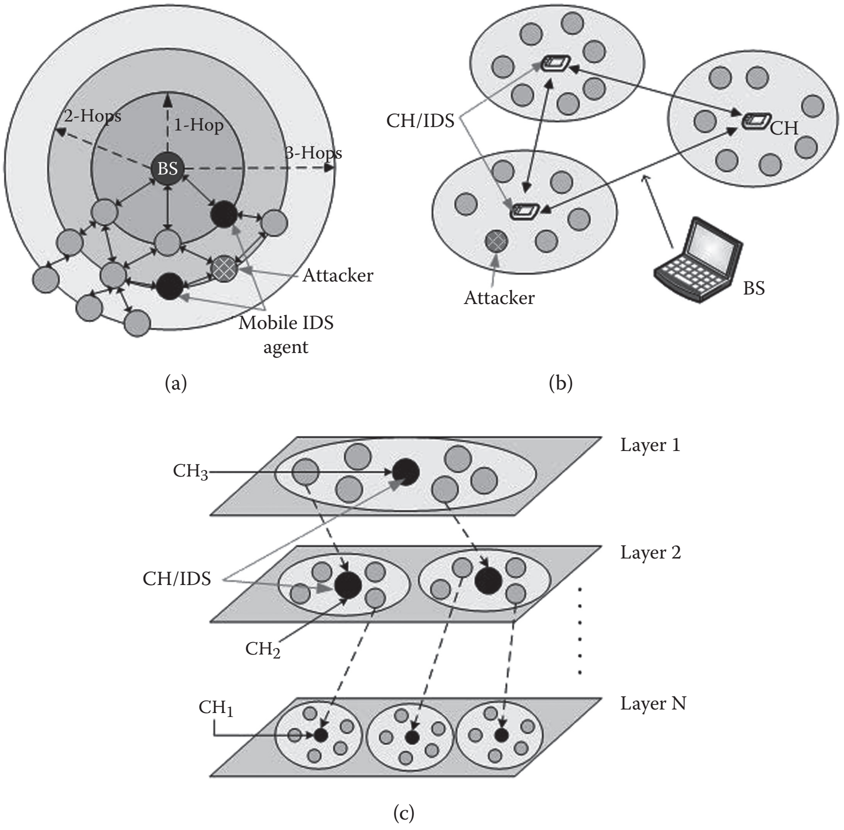


FIGURE 19.8 Three types of network structures with possible IDS locations. (a) Tree-based, (b) Gluster-based and, (c) Hierarchial.

IDS could have several mobile agents in leaf nodes and a global agent in parent node (i.e., base station). This helps the IDS to detect attacks with higher accuracy at the same time maintaining less resource [50].

Furthermore, we believe that it would be very efficient to have one IDS agent for a group of sensor nodes (i.e., installed on a cluster head) in cluster-based network structures. Assuming that cluster heads are slightly more powerful devices than their cluster members, we can implement powerful IDS modules on them (which may not be efficient on typical sensor nodes).

It might be a challenging problem to select the perfect IDS locations for hierarchical structure, which includes both tree-based and cluster-based network structures (thematically). Hence, we advise using a combination of mobile agents between layers and static agents in cluster heads.

19.8 CONCLUSIONS AND FUTURE EXPECTATIONS

In this chapter, we considered a dense deployment scenario of WSNs and have proposed an energy-aware routing protocol that ensures data transmission security for the network. According to our design goals, our protocol structures the network in an energy-efficient way; the base station or the intermediate nodes can detect the presence of falsely injected data, and the network is robust enough to node failures. In this work, in case of security, we have mainly considered the delivery of authenticated and encrypted data from the sensors to the base station. To cover various aspects of security in WSN, alongside presenting our protocol, we offered a comprehensive discussion on the features of security that could be considered for such type of network. Also, we offered a discussion on the usage of an intrusion detection system alongside our proposed mechanism to ensure a complete security architecture for an implemented wireless sensor network with maximum security features maintaining the requirement of energy efficiency.

Other security schemes could be built upon our scheme to protect the network from other types of attacks, if any. In fact, there is a lot of scope to extend the work further. As an example, it could be an interesting topic to find out an optimal value of the time interval for periodic restructuring of the network, so that the maximum longevity of the network could be ensured.

AUTHOR'S BIOGRAPHY

Tarem Ahmed was born in Dhaka, Bangladesh. He received a bachelor's degree with a double major in physics and economics from Middlebury College, Middlebury, VT, USA, in 1999 and a master's degree in electrical engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2000. After serving in industry as an ASIC design engineer in the Silicon Valley area of CA, USA, he has held research positions at the department of electrical and computer engineering at McGill University, Montreal, QC, Canada, and the Computer Engineering and Networks Laboratory at the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. He is presently affiliated with the department of computer science at the International Islamic University Malaysia, Kuala Lumpur, Malaysia, and the department of electrical and electronic engineering at BRAC University in his native city of Dhaka, Bangladesh.

Al-Sakib Khan Pathan received a Ph.D. degree in computer engineering in 2009 from Kyung Hee University, South Korea. He received a B.Sc. degree in computer science and information technology from Islamic University of Technology (IUT), Bangladesh, in 2003. He is currently an assistant professor in the computer science department in International Islamic University Malaysia (IIUM), Malaysia. Until June 2010, he served as an assistant professor in the computer science and engineering department in BRAC University, Bangladesh. Prior to holding this position, he worked as a researcher at Networking Lab, Kyung Hee University, South Korea, till August 2009. His research interests include wireless sensor networks, network security, and e-services technologies. He is a recipient of several awards/best paper awards and has several publications in these areas. He has served as a chair, organizing committee member, and technical program committee member in numerous international conferences/workshops such as HPCS, ICA3PP, IWCMC, VTC, HPCC, IDCS, etc. He is currently serving as the editor-in-chief of *IJIDS*, an area editor of *IJCNIS*, editor of *IJCSE, Inderscience*, associate editor of *IASTED/ACTA Press IJCA and CCS*, guest editor of some special issues of top-ranked journals, and editor/author of nine books. He also serves as a referee of some renowned journals. He is a member of Institute of Electrical and Electronics Engineers (IEEE), USA; IEEE Communications Society, USA; IEEE ComSoc Bangladesh Chapter, and several other international professional organizations.

REFERENCES

1. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. Wireless sensor networks: A survey. *Computer Networks*, 38 (2002), 393–422.
2. Dai, S., Jing, X., and Li, L. Research and analysis on routing protocols for wireless sensor networks. In *Proceedings of the ICCCS*, Volume 1 (27–30 May, 2005), 407–411.
3. Karlof, C., and Wagner, D. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad Hoc Network Journal, SI on Sensor Network Applications and Protocols*, (Sept. 2003), 293–315.
4. Pathan, A.-S. K., Lee, H.-W., and Hong, C. S. Security in wireless sensor networks: Issues and challenges. In *Proc. of IEEE ICACT '06*, Vol. II, Phoenix Park, Korea, (20–22 February 2006), 1043–1048.
5. Çam, H., Özdemir, S., Muthuavinashiappan, D., and Nair, P. Energy efficient security protocol for wireless sensor networks. In *IEEE 58th VTC 2003 Fall*, 2003, 5 (6–9 Oct. 2003), 2981–2984.
6. Çam, H., Özdemir, S., Nair, P., Muthuavinashiappan, D., and Sanli, H. O. Energy-efficient secure pattern based data aggregation for wireless sensor networks. *Com. Commun.*, 29, I.4, (2006), 446–455.
7. Ye, F., Luo, H., Lu, S., and Zhang, L. Statistical en-route filtering of injected false data in sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4), (April 2005), 839–850.
8. Zhu, S., Setia, S., Jajodia, S., and Ning, P. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In *Proceedings of S&P*, (2004), 259–271.
9. Lee, H. Y., and Cho, T. H. Key inheritance-based false data filtering scheme in wireless sensor networks. *Lecture Notes in Computer Science, LNCS 4317*, Springer-Verlag, (2006), 116–127.
10. Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of the 33rd HICSS* (2000), 3005–3014.
11. Azzedine, B., Xiuzhen, C., and Joseph, L. Energy-aware data-centric routing in microsensor networks. In *Proceedings of the 8th MSWiM 03*, San Diego (2003), 42–49.
12. Hyunh, T. T., and Hong, C. S. An energy * delay efficient multi-hop routing scheme for wireless sensor networks. *IEICE Trans. on Information and Systems*, E89-D(5) (May 2006) 1654–1661.
13. Yin, C., Huang, S., Su, P., and Gao, C. Secure routing for large-scale wireless sensor networks. In *Proceedings of IEEE ICCT 2003*, 2 (9–11 April 2003), 1282–1286.
14. Xbow Sensor Networks, Available at: http://www.xbow.com/
15. Hass, Z. J. Design methodologies for adaptive and multimedia networks. *IEEE Communications Magazine*, 39(11), (November 2001), 106–107.
16. Heinzelman, W. B., Chandrakasan, A. P., and Balakrishnan, H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wire. Commun.*, 1(4) (2002), 660–670.
17. Lamport, L. *Constructing digital signatures from one-way function*. Tech. report SRI-CSL-98, 1979.
18. The Network Simulator–ns-2, http://www.isi.edu/nsnam/ns/
19. Coppersmith, D., and Jakobsson, M. Almost optimal hash sequence traversal. In *6th International Financial Cryptography 2002*, Bermuda, (March 2002).
20. Jakobsson, M. Fractal hash sequence representation and traversal. In *2002 IEEE International Symposium on Information Theory*, Switzerland, (July 2002).
21. Sella, Y. On the computation-storage trade-offs of hash chain traversal. In *the 7th International Financial Cryptography Conference*, Guadeloupe, (January 2003).
22. Ee, C. T., and Bajcsy, R. Congestion control and fairness for many-to-one routing in sensor networks. In *Proceedings of ACM SenSys '04*, (2004), 148–161.
23. Pathan, A.-S. K., and Hong, C. S. A secure energy-efficient routing protocol for WSN, *ISPA 2007, LNCS 4742*, Springer-Verlag 2007, 407–418.
24. Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., and Culler, D. E., SPINS: Security protocols for sensor networks, *Wireless Networks*, 8, 2002, 521–534.
25. Xu, W., Trappe, W., Zhang, Y., and Wood, T. The feasibility of launching and detecting jamming attacks in wireless networks, Proc. 6th ACM Int'l. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '05), Urbana-Champaign, IL, May, 2005.

26. Cagalj, M., Capkun, S., and Hubaux, J.-P., Wormhole-based anti jamming techniques in sensor networks, *IEEE Transactions on Mobile Computing*, 6(1), (2007), 100–114.
27. Chen, H., Han, P., Zhou, X., and Gao, C., Lightweight anomaly intrusion detection in wireless sensor networks, PAISI 2007, LNCS 4430, 105–116.
28. Wang, Q., Zhu, Y., and Cheng, L., Reprogramming wireless sensor networks: Challenges and Approaches, *IEEE Network*, May, 2006, 48–55.
29. Pathan, A.-S. K., Lee, H.-W., and Hong, C. S., Security in wireless sensor networks: Issues and challenges, Proceedings of the 8th International Conference on Advanced Communication Technology (IEEE ICACT 2006), Volume II, 20–22 February, Phoenix Park, Korea, 2006, 1043–1048.
30. Krontiris, I., Dimitriou, T., Giannetsos, T., and Mpasoukos, M., Intrusion detection of sinkhole attacks in wireless sensor networks, LNCS, 4837, (2008) 150–161.
31. Krontiris, I., Dimitriou, T., Giannetsos, T., and Mpasoukos, M., Intrusion detection of sinkhole attacks in wireless sensor networks, 3rd International Workshop on Algorithmic Aspects of Wireless Sensor Networks (AlgoSensors '07), Wroclaw, Poland, 2007.
32. Ngai, E. C. H., Liu, J., and Lyu, M. R. An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks, *Computer Commun.*, 30, (2007) 2353–2364.
33. Raymond, D. R., and Midkiff, S. F. Denial of service in wireless sensor network: Attacks and defenses, *IEEE Pervasive Computing*, 7(1), March, 2008, 74–81. [book] Pathan, A.-S. K., *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*, ISBN: 978-1-4398-1919-7, Auerbach Publications, CRC Press, Taylor & Francis Group, USA, 2010.
34. Kaplantzis, S., Shilton, A., Mani, N., Kaplantzis, Y. A. S., Shilton, A., Mani, N., and Sekercioglu, Y. A. Detecting selective forwarding attacks in wireless sensor networks using support vector machines, ISSN IP 2007, Melbourne, Australia, 2007, 335–340.
35. Hai, T. H., and Huh, E. N. Detecting selective forwarding attacks in wireless sensor networks using two-hops neighbor knowledge, In Proc. of the 2008 Seventh IEEE International Symposium on Network Computing and Applications, 2008, 325–331.
36. Demirbas, M., and Song, Y. An RSSI-based scheme for sybil attack detection in wireless sensor networks, In Proc of IEEE WoWMoM, 2006, 564–570.
37. Loo, C. E., Ng, M. Y., Leckie, C., and Palaniswami, M. Intrusion detection for routing attacks in sensor networks, *International Journal of Distributed Sensor Networks*, 2(4), (2006), 313–332.
38. Zhou, J., Das, T. K., and Lopez, J. An asynchronous node replication attack in wireless sensor networks, Proceedings of the IFIP TC 11 23rd International Information Security Conference, 278, Boston Springer, 2008, 125–139.
39. Parno, B., Perrig, A., and Gligor, V. Distributed detection of node replication attack in sensor networks, IEEE S&P, 2005.
40. Wang, Y., Attebury, G., and Ramamurthy, B. A survey of security issues in wireless sensor networks, *IEEE Communications Surveys & Tutorials*, 8(2) 2nd Quarter, 2006.
41. Hamid, M. A., Mamun-Or-Rashid, M., and Hong, C. S. Routing security in sensor network: HELLO flood attack and defense, Proceedings of IEEE ICNEWS 2006, Dhaka, Bangladesh, 2–4 January 2006, 77–81.
42. Sharif, W., and Leckie, C. New variants of wormhole attacks for sensor networks, In Proc. of the Australian Telecommunication Networks and Applications Conference, 2006, 26–30.
43. Hu, C. Y., and Perrig, A., Wormhole attacks in wireless networks, *IEEE Journal on Selected Areas in Communications*, 24(2), (2006), 370–380.
44. Maheshwari, R., Gao, J., and Das, S. R., Detecting wormhole attacks in wireless sensor networks using connectivity information, In Proc. of INFOCOM 2007, 107–115.
45. Graaf, R. D., Hegazy, I., Horton, J., and Safavi-Naini, R. Distributed detection of wormhole attacks in wireless sensor networks, Ad Hoc Networks, LNCS, 28(1), (2010) 208–223.
46. Newsome, J., Shi, E., Song, D., and Perrig, A. The sybil attack in sensor networks: Analysis & defense, In Proc. of ACM IPSN '04, 2004, 259–268.
47. Mukhopadhyay, D., and Saha, I. Location verification based defense against sybil attack in sensor networks, ICDCN 2006. LNCS 4308, Springer-Verlag 2006, 509–521.
48. Chen, R. C., Haung, Y. F., and Hsieh, C. F. Ranger intrusion detection system for wireless sensor networks with Sybil attack based on ontology, AIC '10, 2010.
49. Pathan, A.-S. K. Security of self-organizing networks: MANET, WSN, WMN, VANET., ISBN: 978-1-4398-1919-7, Auerbach Publications, CRC Press, Taylor & Francis Group, USA, 2010.
50. Roman, R., Zhou, J., and Lopez, J.- Applying intrusion detection systems to wireless sensor networks, Consumer Communications and Networking Conference, 1, 2006, 640–644.