PET ENGINEERING COLLEGE

An ISO 9001:2015 Certified Institution

Accredited by NAAC, Approved by AICTE, Recognized by Government of Tamil Nadu and Affiliated to Anna University

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

UNIT - IV

SENSOR NETWORKS SECURITY

CLASS : S7 ECE

SUBJECT CODE : EC8702

SUBJECT NAME : ADHOC AND WIRELESS SENSOR NETWORKS

REGULATION : 2017

further references are [142, 907]. Reference [457] also considers heterogeneous networks (i.e. networks with more powerful nodes added) for such an application.

Compression and routing Reference [728] combines the concepts of network coding and routing. They characterize conditions for the rate/distortion function to be applied so that all nodes can broadcast their readings in the entire network, given a prescribed quantization error.

In a somewhat similar context, PATTEM et al. [623] look at the performance of routing in combination with compression ad aggregation in presence of spatial correlation of the observed data. They claim a static clustering scheme achieves near-optimal performance.

Transport capacity and measurement accuracy MARCO et al. [538] address the principal question how much data needs to be extracted from a measurement field to attain a given measurement error and how this number relates to the measurement accuracy of the sensor network. Moreover, they study how the number of sensors influences these results.

14.2 Security

Network security [729, 731, 785] is one of the most pressing concerns in all wireless networks, including wireless sensor networks. In this section, we briefly introduce the security problem and explain some of the specifics of wireless sensor networks. The discussion is in parts based on SCHÄFER [729, 730].

14.2.1 Fundamentals

Network designers have to be aware of and decide about suitable mechanisms to implement one or more of the following general **security goals** [729, Sec. 1.2]:

Confidentiality Information should only be revealed to authorized entities; any other entity should not be able to discover the information from eavesdropping or from reading memories.

Data integrity The receiver of information wants to be sure that it is not modified in transit, either intentionally or by accident. To distinguish unmodified "wanted" information from unmodified bogus information, the originator must be identifiable uniquely.

Accountability The entity requesting a service, triggering an action, or sending a packet must be uniquely identifiable.

Availability Legitimate entities should be able to access a certain service/information and to enjoy proper operation.

Controlled access A service or information access should only be granted to authorized entities.

Any security analysis must start with stating the desired security goals, followed by an assessment of the possible risks or security threats posed by an attacker. Some common threats are eavesdropping, masquerading (i.e. pretending to have another entity's identity), authorization violation (using services without being allowed to use them), provoking loss or modification of information, forgery (i.e. creating new information), repudiation, and sabotage.

When considering networking, some of the common attacks are eavesdropping as a purely passive attack, and insertion, deletion, or replaying of packets as an active attack. Attacks can be placed on all the layers of a given protocol stack.

Many countermeasures have been developed against these threats. These mechanisms frequently rely on symmetric or asymmetric **cryptographic algorithms** [731], [548]. These algorithms can be used to encrypt data packets, to sign these with almost unique hash/cryptographic check values, or to create certificates. Cryptographic algorithms essentially work by applying certain operations on combinations of the user data and specific **key values**, which optimally are only known to the sender and the receiver of a packet. Distributing these keys to the users and taking care of their lifecycle are essential parts of **key management** protocols. In practice, key management turns out to be the most complex part of security protocols; the raw encryption and decryption procedures are small but important building blocks.

14.2.2 Security considerations in wireless sensor networks

Can security measures and cryptographic protocols in wireless sensor networks be considered in the same way as for other types of networks? There is some consensus that the answer seems to be "no", for the following reasons:

- The network infrastructure of a WSN is made up of small, cheap nodes spread over a possibly hostile area. Unlike other types of networks, it is often impossible to prevent the sensor nodes from being physically accessed by attackers. This is also referred to as **node capture**. It is reasonable to assume that an attacker can achieve full control over a captured node, that is he can read its memory or influence the operation of the node software. Special secure memory devices would be needed to prevent the attacker from reading the memory; however, these will only rarely be present in cheap sensor nodes.
- The constraints regarding memory and computational capabilities are a serious obstacle for implementing cryptographic algorithms. Especially asymmetric key cryptography is considered too heavyweight for small processors, let alone the key management involved. The usage of several cryptographic block ciphers in sensor networks has been investigated in reference [471].
- When in-network processing is to be performed, intermediate nodes need to access and modify the information contained in packets; hence, a larger number of parties is involved in end-to-end information transfers.
- The finite energy budget of sensor nodes opens up a particularly attractive line of attacks: to force victim sensor nodes to exhaust their energy budget quickly and to die.

An additional challenge pointed out by Schäfer [730] is that attackers can have much more energy at their disposal than the sensor nodes. All security measures carried out by a sensor node require extra energy and stressing the node by attacks can cause premature depletion. This amounts to one particular kind of a **denial-of-service** attack (DoS). In the following, some of these DoS attacks are briefly described.

14.2.3 Denial-of-service attacks

Wood and Stankovic [891] consider a number of different denial-of-service attacks in sensor networks, working at different levels. Denial-of-service attacks in general can try to [730] (i) disable services, or (ii) to deplete service providers, for example, by overusing the service. To disable a sensor network's service, an attacker might simply destroy nodes. Although sensor networks have some resilience to node failures, the attacker can distort the network by destroying a large number of nodes or by focusing on especially important nodes, for example, sensor nodes in the vicinity of sinks that are needed for forwarding. In the following, however, we discuss protocol-related attacks.

Physical-layer and link-layer attacks

With **physical-layer jamming**, an attacker simply distorts radio communication. One way to achieve this is to place attacker nodes somewhere into the network and let them continuously send radio signals in the sensor network's frequency band. Especially effective is such an attack when the attacker nodes are close to sink nodes, effectively reducing a user's ability to control the network or to acquire data from it. A single attacker node can distort many neighbors at once and, by strategical placement of a number of attacker nodes, the whole sensor network can be disabled.

One possible countermeasure is the use of modulation schemes with some robustness against interference, for example, frequency-hopping or direct-sequence spread-spectrum techniques ([293, 297, 557]; see also Section 4.2.5). A second possible countermeasure is that the uncompromised sensor nodes reduce their duty cycle upon detecting such an attack. If the attacker has itself only a finite energy budget, it can persevere only for a limited time. A third countermeasure can be taken by routing protocols: If the attacker jams only a limited area, packets may be routed around. In protocols like directed diffusion, frequent interest dissemination can find working routes. Finally, sensor nodes with different physical layers can switch between these (for example, between a radio and an infrared transceiver).

A cleverer attacker can take knowledge about the protocols into account to save energy, giving rise to **link-layer jamming**. Especially, the MAC protocol is a good candidate. Let us consider, for example, protocols based on exchange of RTS/CTS packets (see Section 5.1.2) like PAMAS (Section 5.3.2) or S-MAC (Section 5.2.2). Whenever an attacker node a receives an RTS packet issued by some node x, it can answer with a jamming signal, interfering with any CTS packet sent to x. As a consequence, x has no transmit opportunity, backs off and tries again later with another RTS packet. According to Wood and Stankovic [891] no effective countermeasure against such an attack exists. The attacker might exploit the MAC protocol further to save energy. For example, in S-MAC the attacker can adapt its activity periods to the schedules of its neighbors.

Another ugly attack exploits MAC protocols using immediate acknowledgments and retransmissions. Upon receiving a data frame from node x, the attacker node can jam the acknowledgement frame destined to x. This causes x to back off, retransmit the same packet and to waste energy. Another way of depleting a node x is to continuously send RTS packets to this node, causing him to answer with CTS packets.

Network-layer attacks

Several types of attacks can be executed on the network layer. First, attacker nodes can behave similar to normal nodes; specifically, they can participate in routing protocols or dissemination of interests with the goal of directing routes to itself and to drop packets later on. This attack is called **black hole** attack. For example, in distance-vector protocols, the attacker can pretend to have particularly good routes to the sink. Dropping of packets destroys information, and furthermore, the forged route advertisements attract lots of traffic around the attacker, causing increased congestion levels and contention.

In a similar kind of attack, so-called **misdirections**, the ~~adversary~~ attacker creates wrong routes, for example, by sending wrong route advertisement packets or by falsely answering route request packets. A wrong route can, for example, contain a loop and cause waste of energy. Another possible effect is that traffic does not reach the intended sink nodes. Instead of creating wrong routes, an adversary can also cause creation of unnecessary routes, for example, by issuing route lookup requests. All nodes participating in route selection waste their energy.

Even without actively trying to be included as a forwarder into routes, an attacker node can drop other nodes' packets and forward only its own packets. Such an attack is called **neglect and greed**. The attacker node can drop packets in a random fashion or all of them. Routing or data dissemination protocols that cache routes (like DSR or directed diffusion) are vulnerable to this attack. The attacker node participates in route setup and distorts, later on, the forwarding of data packets. When this behavior has been detected, the network may set up alternate routes or a source node can send multiple copies of a packet over node-disjoint routes from the beginning.

All these attacks have their source in adversary nodes participating in routing protocols. To prevent this, authentication and/or authorization mechanisms are needed to restrict routing protocols only to trustworthy nodes. Protocols for this purpose are beyond the scope of this chapter.

An attack called **homing** seeks to determine the geographic locations of certain important nodes in the network, for example clusterhead nodes. This information can be obtained from eavesdropping location-centric protocols. Once this information has been determined, the adversary can direct other attacks to these nodes. Clearly, a good way to prevent this attack is encryption of location information.

Transport layer and application attacks

If the transport layer uses explicit connections between identifiable nodes, either end of the connection needs to maintain some form of connection control block (CCB). Similar to TCP syn flood attacks, an attacker can issue a large number of connection setup requests and cause exhaustion of memory at the end nodes because of large numbers of unneeded CCBs.

Another kind of attack identified by Wood and Stankovic [891] is **desynchronization**, which can be applied to transport protocols resting on sequence numbers. By issuing forged packets with wrong sequence numbers, the attacker can cause wasteful retransmissions or even cause the participants to end the connection.

In sensor networks deployed to detect certain environmental events, an attacker node can generate sensor data indicating this event, causing nodes in the vicinity or even the whole network to wake up and to start various activities. Possible countermeasures can be developed starting from outlier detection techniques.

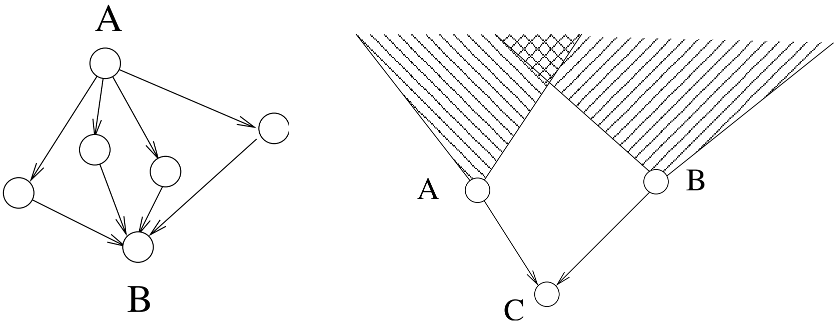
14.2.4 Further reading

- Reference [730] is a survey article on sensor network security. It discusses some key management protocols in further detail. Key management issues in sensor networks are also discussed by Eschenauer and Gligor [244] and Zhu et al. [937].
- Karlof and Wagner [406] consider secure routing and network-layer attacks in sensor networks in some more detail and discuss also a number of countermeasures.
- Perrig et al. [638] consider among others the problem of secure broadcast; see also [637].
- The *ACM Conference on Computer and Communication Security (CCS)* and the *IEEE Symposium on Security and Privacy* regularly present papers on sensor network security issues.

14.3 Application-specific support

In this section, we briefly describe three different tasks that sensor networks might be tasked with and which likely are important building blocks of sensor network applications. The first one is detection and tracking of (mobile) targets, for example, intruders into some site, the second one is detection of edges or of contours/isolines in the level of a continuous physical phenomenon, and the third is to obtain an estimate of a physical field. We also sketch some ideas of how these tasks can be approached with sensor networks.

All these applications are good showcases for geographic forwarding and geographic addressing since they make heavy use of position information and incorporate this into forwarding and routing decisions; this is especially true for tracking.



(a) Implosion (b) Data overlap

Figure 7.4 Main problems with flooding.

Although gossiping avoids the implosion problem by just having one copy of a message at any node, it increases the latency in propagating the message to all sensor nodes. Since a single node at a time is informed about the packet, the information is distributed slowly. On the other hand, since multiple copies of a packet are prevented, the energy consumption of gossiping is lower than that of flooding.

Although simple and inefficient, flooding and/or gossiping techniques can still be used by recent routing protocols for specific functions. As an example, during the deployment phase, the sink can use flooding or gossiping protocols to determine the active nodes. Similarly, during sensor network initialization, limited flooding can be used to gather information from neighbors in close proximity.

7.2.3 Sensor Protocols for Information via Negotiation (SPIN)

SPIN is a family of routing protocols designed to address the deficiencies of flooding by negotiation and resource adaptation [13]. For this purpose, two main approaches are followed. Firstly, instead of sending all the data, sensor nodes negotiate with each other through packets that describe the data. Consequently, the observed information is only sent to interested sensor nodes as a result of this negotiation. Secondly, each node monitors its energy resource, which is used to perform *energy-aware* decisions.

■ **EXAMPLE 7.2**

The negotiation mechanism of SPIN is performed through three types of messages, namely advertisement (ADV), request (REQ), and DATA, which are illustrated in Figure 7.5. Before sending a DATA packet, a node advertises its intent by broadcasting an ADV packet (Step 1). The ADV packet contains a description of the DATA packet to be sent, which is much smaller in size than the DATA packet. Then, if a neighbor is interested in the ADV packet, it replies back with a REQ message (Step 2). Finally, the DATA packet is sent to the node that requests it (Step 3). Data propagation in WSNs is coordinated through this mechanism at each hop. As shown in Steps 4, 5, and 6 of Figure 7.5, multiple nodes can send REQ messages back to a node, which sends DATA to each node until all the nodes get a copy. As a result of the SPIN protocol, the sensor nodes in the entire sensor network which are interested in the data will get a copy.

The basic operation explained in Example 7.2 is referred to as the point-to-point SPIN protocol (SPIN-PP). In addition to SPIN-PP, several variations have been proposed to address some of the disadvantages of SPIN-PP. We explain these variations of SPIN next, i.e., SPIN with energy consumption awareness (SPIN-EC), SPIN for broadcast networks (SPIN-BC), and SPIN with reliability (SPIN-RL).

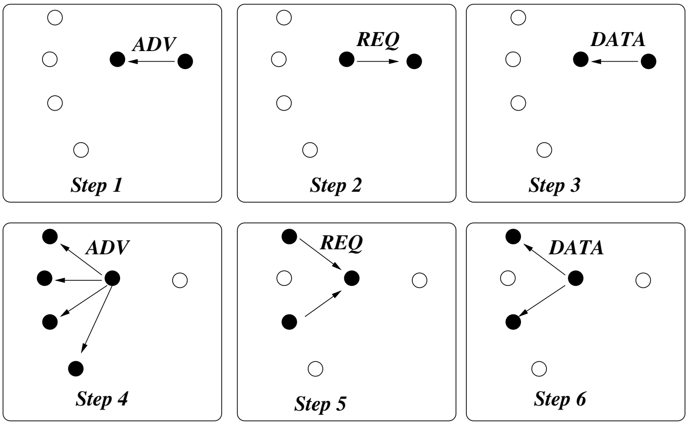


Figure 7.5 The SPIN protocol [13].

SPIN-PP does not address the resource-blindness problem of conventional flooding or gossiping protocols. Although the DATA packet transmission is limited to nodes that provide interest, energy consumption is still a concern. SPIN-EC addresses this through a simple energy conservation heuristic such that whenever the residual energy of a node is lower than a threshold, the node does not participate in the protocol operation, i.e., it does not send a REQ packet if it does not have enough energy to transmit the REQ packet and receive a DATA packet. Since node participation is dependent on the residual energy, if a node has plenty of energy SPIN-EC behaves like SPIN-PP.

Another disadvantage of SPIN-PP is shown in Steps 5 and 6 of Figure 7.5. Whenever there is more than one node that sends REQ packets, the DATA packet is sent to each node individually. Considering the broadcast nature of the wireless channel, this approach is a waste of resources since each neighbor of a node can receive the packet in each unicast. Furthermore, SPIN-PP does not provide any mechanism to prevent collisions when multiple REQ packets are send. This is addressed through SPIN-BC, which is developed for broadcast networks. In contrast to SPIN-PP, SPIN-BC introduces a randomized backoff mechanism for the nodes before transmitting a REQ packet. As a result, if a node has an interest in a packet but hears a REQ packet related to that particular packet, it drops its REQ packet and waits for the DATA packet. Upon receiving a REQ packet, a transmitter node broadcasts a single DATA packet which can be received by all the interested neighbors. As a result, SPIN-BC decreases the energy consumption and overhead caused by multiple interested neighbors.

SPIN-RL provides a reliability mechanism to the SPIN-BC protocol such that if a node receives an ADV packet but does not receive a DATA packet followed by it (due to wireless channel errors), it requests the DATA packet from the neighbors that may have received the DATA packet. Moreover, SPIN-RL limits the retransmission period of the nodes such that they do not retransmit a DATA packet before a specified period.

SPIN is based on data-centric routing [13] where the sensor nodes broadcast an advertisement for the available data and wait for a request from interested sinks. Compared to flooding, SPIN-PP reduces energy consumption by 70% since redundant transmissions are prevented. SPIN-EC provides a further 10% increase in energy consumption, through energy-aware operation. Moreover, since local interactions are required for routing, SPIN is also scalable. However, compared to flooding, the latency in data dissemination is higher because of the overhead in the handshake mechanism.

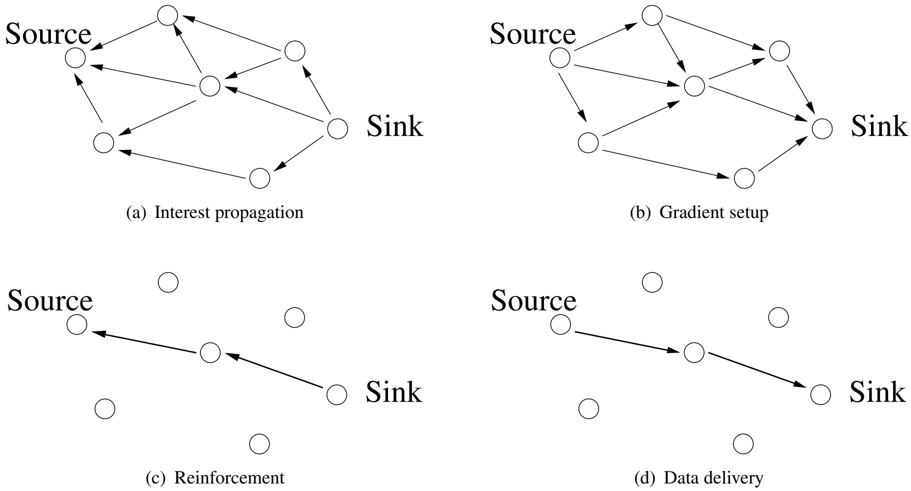


Figure 7.6 Operation of the directed diffusion protocol [14].

7.2.4 Directed Diffusion

SPIN provides efficient mechanisms for a sensor node to disseminate its observations to interested nodes. As a result, the traffic flow in SPIN is initiated from the sensors and usually ends up at the sink. However, this type of traffic may not always be preferable when the user (i.e., sink) requests specific information from the sensors. The *directed diffusion* data dissemination paradigm has been developed to address this requirement [14]. Directed diffusion consists of four stages to construct routes between the sink and the sensors of interest to the sink's request. The four stages are: (1) interest propagation, (2) gradient setup, (3) reinforcement, and (4) data delivery, as described next.

The information request is provided through *interest* messages initiated by the sink. Directed diffusion is initiated when the sink sends out interest messages to all sensors as shown in Figure 7.6(a). This phase is called *interest propagation*, where the interest messages flood through the network. The interest messages act as *exploratory* messages to indicate the nodes with matching data for the particular task. During the task, the sink continues to periodically broadcast the interest message.

Upon receiving the interest message, each sensor node stores it in an interest cache. The interest cache has several fields including *timestamp*, *gradient*, *interval*, and *duration*. The timestamp field indicates the local time when the interest is received. The gradient indicates the node from which the interest has been received. This gradient field is used to form reverse paths towards the sink. Each interest is stored at the cache for a specific time indicated by the duration field. After receiving the interest, the sensor node forwards the message to its downstream neighbors. This forwarding can be similar to flooding or more limited according to the task description. Local rules can be specified to define different gradient setup techniques. Accordingly, the gradient can be sent to the node which first sends the interest. Similarly, the gradient can be set up such that nodes with the highest remaining energy are chosen. As the interest is propagated throughout the sensor network, the gradients from the source back to the sink are set up as shown in Figure 7.6(b).

The interest messages indicate the required data at a given time from the sensor networks. Each node checks its sensor observations and becomes a source node if it has data matching the interest. When the source has data for the interest, the source sends the data along the interest's gradient path as shown in Figure 7.6(b).