Chapter 14
Wireless Sensor Nodes

Serge Chaumette and Damien Sauveron

Abstract This chapter addresses the key points of wireless sensor nodes: applications, constraints, architecture, operating systems, and security concerns. It does not pretend to be exhaustive but to provide the major references on these topics.

14.1 Introduction

Huge advances in Microelectromechanical systems (MEMS) and Wireless communications in the last decade of the twentieth century gave birth to new paradigms, where cheap, small size communicating sensors have been developed and integrated in many devices and large hardware/software environments. In this chapter, we target standalone wireless sensing devices, so-called wireless sensor nodes, which means that we focus on the device itself and not on the way it can be integrated within a global wireless sensor network. These small hardware pieces are becoming a key component of the digitization of the real world, thanks to their ease of deployment and the benefits that they can bring to human life in general, like infrastructure management (such as power grids) and environmental protection for instance. They are quite different from expensive isolated sensors (i.e., not intended to be a part of a whole swarm) and achieve complex measurements (related to a given phenomenon), and subsequent computation operations. Indeed, the strength of small sensor nodes is their ability to self-organize as a large network which enables measurement very close to a possibly dangerous phenomenon. They can cover a wide area and so are able to observe the evolution and spreading of complex events. In Sect. 14.2 we

S. Chaumette (✉)
LaBRI,University of Bordeaux, Bordeaux, France
e-mail: serge.chaumette@labri.fr

D. Sauveron
University of Limoges, Limoges, France
e-mail: damien.sauveron@unilim.fr

K. Markantonakis and K. Mayes (eds.), *Secure Smart Embedded Devices, Platforms and Applications*, DOI: 10.1007/978-1-4614-7915-4_14,

will show that wireless sensor nodes are used in both military and civil applications, health care, and power grid monitoring, among many other domains. In Sect. 14.3 we address constraints, like cost, energy-consumption, and network management (even though, as explained above, this last point is not a central topic of this chapter). In Sect. 14.4 the generic architecture of a wireless sensor node and the major features of the associated operating systems are described. We eventually present the security concerns in Sect. 14.5.

14.2 Applications

It is widely acknowledged that the first uses of advanced sensing technologies were military. In 1949, the US Navy announced its intention to exploit passive sonars for anti-submarine warfare (ASW) purposes using hydrophonic sensors deployed on the seafloor. In 1961, the sound surveillance system (SOSUS) [4] provided deep-water long-range detection capabilities, which were successfully used during the Cold War. It is now used by national oceanographic and atmospheric administration (NOAA) to observe a number of natural phenomena like submarine earthquakes [51] or the activities of some animals [22]. However, the concept of Distributed Sensor Networks only appeared in the early 1980's, driven by defense advanced research projects agency (DARPA) through a first program which was followed in 2001, due to evolution in the domain of MEMS, by the sensor information technology (SensIT) program [43]. The primary goals of SensIT were the creation of a new class of software for distributed microsensors, the development of novel methods for ad hoc networking of deployable microsensors and the extraction of right and timely information from a sensor field. Typical military applications of sensors are related to the collection of information that can be used to enforce situation management:

- monitoring friendly forces, equipment, and ammunition;
- battlefield surveillance;
- reconnaissance of opposing forces and terrain;
- targeting (e.g., for missiles);
- battle damage assessment;
- nuclear, biological, and chemical attack detection and reconnaissance.

Since early 2000, the number of civil applications has increased. Sensors now permit the creation of low-cost monitoring systems. Many domains, for instance biology, climatology, geology, etc., benefit from integrated sensors and sensor network technology. These sensors can for instance be used:

- to alert the civilian population when a volcanic eruption is going to occur [67] and to collect data without any risk to compromise the physical integrity of the persons in charge of taking the samples [65];
- to detect forest fire [37] or floodings [15];
- to forecast environmental pollution;

- to observe animals in their living environment without human presence [42, 49];
- to help farmers collect accurate measures of phenomena, which have a potential impact on their agricultural production [16].

As mentioned in the introduction, sensors can also be used in:

- health care, like telemonitoring of human physiological data [41], tracking and monitoring doctors and patients inside a hospital, environmental control in office buildings [54], etc.;
- infrastructure protection by monitoring bridges, tunnels, pipelines, power grids, etc.;
- home automation and smart environments [32], interactive museums [54], vehicle tracking and detection [60], etc.

Beyond these well-known applications, some other are much more unusual. For example, Simon et al. [61] propose to use a large number of cheap sensors communicating through an ad hoc wireless network, to detect and accurately locate shooters in urban environments. Their system supports multiple sensors failures, provides good coverage and high accuracy, and is capable of overcoming multipath effects.

14.3 Constraints

As illustrated in the above section, Wireless Sensor Nodes can be used for an extremely wide range of applications. However, they suffer from several constraints that need to be known and understood in order to properly target their possible domains of use.

14.3.1 Costs: Production Versus Performance

When a phenomenon needs to be observed, the decision to use wireless sensor nodes or classical sensors is mostly based on technical reasons. For instance, when considering the ability of the system to observe the spreading and evolution of a phenomenon to measure, it appears that a wireless sensor network is probably a good candidate because it allows wider and more accurate geographic coverage. Nevertheless, the financial cost of building the sensing system is also a prominent parameter. A wireless sensor network can be composed of a few to several thousands[1] devices and the overall price of the sensing system may thus be huge. To be viable, their production

[1] In practice very few applications use thousands of nodes. However theoretically, for instance, in military applications like battlefield surveillance, a huge number of nodes may be required. The biggest wireless sensor networks publicly known that have ever been built are: a system based on MyriaNed and composed of more than 1000 nodes [14, 38]; an 800 nodes network called the "Largest Tiny Network Yet" deployed at Berkeley [5] in 2001; WISEBED [21] which is made of

cost should remain very low (for example less than 1$/node). Indeed, even if not all networks are composed of thousands of nodes, there can be several networks operating in parallel for different purposes and thus the total number of nodes can still be huge. At the same time, they should offer good performance to sense phenomena and if required enough computing power to handle data locally, so as to overcome the network-related constraints and usage consequences such as energy-consumption (see below). Thus, their design is always the result of a trade-off between performance and cost. Obviously there is no hidden cost due to the deployment of wires (and associated devices) as can be found in classical networks, since by definition wireless sensor nodes do not require any infrastructure, however, in some cases there may be a cost for a wireless sensors spectrum licence.

14.3.2 Energy

To enable the sensors to operate during their whole mission time, energy saving is a major concern of both the manufacturers and the users, even though it is widely acknowledged that designing energy efficient communication components is a challenging task. Furthermore, it still remains that even with a power-efficient radio frequency (RF) technology, the energy consumption due to the communication between two nodes follows at best an inverse-square law of their distances. For this reason, to save power, it is needed to 'split' long distances between two communicating nodes by using multi-hop communication and routing [24]. Software is thus also largely involved in the energy-saving process. However, this way to communicate implies more cooperation between the nodes of the network in order to relay messages using power-efficient routing algorithms. It is thus a challenge to design low duty cycle radio circuits to relay messages between neighbors without losing any message. Strategies based on "wakeup on demand" (using two radios) [59] or "adaptive duty cycles" [68] are being studied to circumvent this problem. Another solution is to setup a network backbone with a subset of nodes that remain active.

In addition, it is possible to minimize [8] power consumption in a multi-hop sensor network, by processing sensed data locally to minimize transmission. The economy comes from the fact that the cost to transmit a large quantity of raw data coming from the sensor is more important than the cost of locally processing them, extracting the useful information, and eventually sending these information over the wireless network. Even if it is true that communication consumes a large quantity of energy in a sensor architecture, it still remains that the rest of the hardware and especially the processing unit must also be designed to be efficient. At software level, the operating system, protocols, algorithms, etc., must also be power-efficient and thus power-aware.

(Footnote 1 continued)
around 750 nodes but split in several subnetworks; the architecture described in [7] that uses 273 sensors and 47 wireless nodes to monitor nectarine orchard.

When battery capacity cannot be sufficient, solar cells [20] can be added to supply additional power. But they increase the manufacturing cost and are not suited to all deployments. However, other power scavenging methods, which enable wireless nodes to be completely self-sustained exist [55], at least as prototypes. Among these methods are those using temperature gradients, human power, wind or air flow, vibrations [56], etc.

14.3.3 Management: Self and Decentralized

When large-scale wireless sensor networks are deployed, relying on a centralized base station that would manage the topology and the routing is by nature not possible. This is due, among other parameters, to the cost in terms of energy spent to communicate over large distances. However, even though it is true that lower cost (in terms of energy consumption) multi-hop communication can be used to reach a base station, it still remains that in the particular case where nodes are mobile, the base station can become physically unreachable. Thus, wireless sensor networks often rely on decentralized management. For instance the decisions related to routing are computed locally based on information collected from neighbors using algorithms that once again optimize the energy consumption of the system.

In addition, it must be remembered that the nodes are deployed in a possibly adverse environment where they must thus operate without any human intervention. Therefore, configuration, adaptation, maintenance, and repair should be performed in an autonomous manner [24]. The nodes then need to have self-management and context-awareness capabilities [50], like self-organization and self healing. These require the ability to adapt configuration parameters according to changes in the environment, such as network disruption (the goal here being to maintain a given network topology) and to support self-protection (the ability to detect and protect against attacks). These are two fundamental features.

14.4 Architecture and Operating System

There are a plethora of readily available wireless sensor nodes which combine diverse hardware architectures with various operating systems. To try and make things clear, this section describes a tentative generic architecture of a node and the main features that are usually supported by the associated operating systems. Figure 14.1 presents the architecture. The sensing unit is the aim for which the sensor node has been designed. The rest of the unit is needed: to process the acquired data items or those received from neighbor nodes; to communicate with neighbor nodes; to supply power to all the boards and components. As can be seen, the sensing, power, and communication units exchange/communicate over buses such as general purpose input/output (GPIO), secure data input/output (SDIO), universal serial bus (USB),

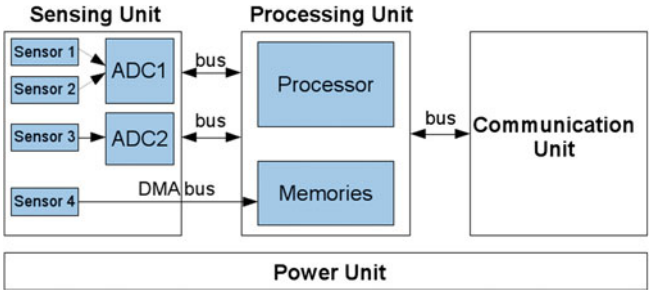

Fig. 14.1 Generic architecture of a wireless sensor node

serial peripheral interface (SPI), Inter-Integrated Circuit (I2C), the two latter being the most commonly used. As illustrated in Fig. 14.1, a direct memory access (DMA) bus can even be used so as to save the computing cycles of the processor.

As aforementioned, energy supply and management are crucial in sensors. The power unit may be a simple battery or it can generate part (or all) of the necessary power using solar cells, or even MEMS-based advanced technologies.

14.4.1 Sensing Unit

Figure 14.1 shows that a sensing unit can integrate several sensors and analog to digital converters (ADCs). The crucial hardware element of a sensor node are obviously the sensors themselves (even though the other components are also of major importance). Their goal is to monitor changes of some external physical phenomenon and to output a continuous analog signal as a function of time, which then needs to be converted to a digital value. The choice of the embedded sensors is done by the manufacturer depending on the physical phenomena to observe. It should also be noted that many architectures now support the notion of shields that are purpose-specific sensor boards, that can be plugged in depending on the target application. Thanks to the advances in MEMS, there exist sensors for most physical values, from heart rate to temperature (an illustrative list of sensors is given Table 14.1). The readers interested in building their own node with specific sensors are referred to a later chapter where the Arduino platform (a Do It Yourself open hardware platform) is described. A sensor that requires power to measure a physical information is called active while one that derives power from the energy provided by the sensed phenomenon is called passive.

To convert the signal produced by a sensor to information that can be handled by the processing unit, an ADC must be used to digitize the analog values. A bus is then used to feed the processing unit with the resulting digital information. As can be seen in Fig. 14.1, an ADC can be shared by different sensors. The decision of having one single ADC for several sensor units or one ADC per sensor unit or even not to embed

Table 14.1 Examples of sensed phenomena and associated sensors

Type of sensed phenomena	Example of sensor
Acoustic	Microphone
Chemical	pH sensor
Electromagnetic	Magnetometer
Flow	Anemometer
Humidity	Hygrometer
Mechanical	Tactile sensor
Motion	Accelerometer
Optical	Photodiode
Position	Gps
Pressure	Barometer
Radiation	Radioactivity counter
Temperature	Thermocouple

any ADC is done by the manufacturer based on the nature of the information (for instance, in terms of sampling frequencies and in terms of resolution) that the sensor is designed to measure. Multiplexing at ADC level can be used for sensors whose monitored information does not change with high frequency (or at least remains under a given threshold).

Although this will not be detailed here, sensor nodes can also integrate actuators such as LEDs, motors, etc.

14.4.2 Processing Unit

The processing unit provides the computing power of the system and connects the sensing and the communication modules. It consists in a processor and different kinds of memory banks: a nonvolatile memory (for instance Flash memory or EEPROM) to store the programs (application(s) and OS) and a volatile fast memory (different brands of RAM are possible) to store the objects used by the operating system (like the application stack), the sensed data items, etc. The processor can be of different types: field programmable gate array (FPGA), digital signal processor (DSP), application-specific integrated circuit (ASIC), or microcontroller. The latter embeds memories and several additional hardware features such as a clock generator.

However, if DSPs and FPGAs are special purpose energy-efficient processors that can be used for some well-defined and simple sensing tasks, they are not suitable for setting up a modular architecture. In the sensor network context, the sensing tasks can vary a lot, the overall hardware configuration can be modified by adding a new sensor, and the software layers must then be able to redefine the sensing operations. It is acknowledged that a microcontroller is the best solution when dynamic code loading and updating are required. For instance, it makes it possible to load post-issuance new energy-efficient software components and protocols. It is less powerful

than a DSP or a FPGA, but easier to program (in C or assembly language). The main drawback of DSPs is their lack of flexibility when reprogramming is needed, whereas the drawback of an FPGA is that its production and design costs remain high. An ASIC provides the best performance, but again reconfiguration is difficult to achieve. Most of the time ASICs are used to complement microcontrollers or DSPs in low level tasks, such as wireless communication.

From an architectural point of view, the processing unit can be organized according to the Von Neumann (for FPGAs and ASICs) or Harvard (or even Super Harvard) architectures (for DSPs or microcontrollers).

14.4.3 Communication Unit

The communication unit, i.e., the transceiver, may be an optical device like in the Smart Dust motes project [50], but most of the time, it is an RF device. However, as aforementioned in Sect. 14.3.2, hardware is not sufficient and software multi-hopping is the most efficient and usual way to communicate in sensor networks. Therefore, different types of short-range and energy-efficient radio technologies are used such as those based on IEEE 802.15.4 like ZigBee [29], WirelessHART [62], ISA100.11a [64]; IEEE 802.11 [23] like WiFi; IEEE 1451 [45]; or other proprietary technologies. Because of its high-power consumption when switching the RF module on and off, Bluetooth is not used very often. Although there are not many benchmarks available, the reader interested in a comparison between these technologies can refer to the three following papers [46, 48, 63].

14.4.4 Major Features of Operating Systems

The operating system of a WSN is a small software layer linking the application layer to the hardware layer. It enables applications to interact with the hardware resources through services or libraries and is in charge of scheduling and prioritizing tasks. It can manage memory, power, etc. In addition, it provides application developers with a few basic abstractions and paradigms to express their algorithms. For example, to support concurrent tasks, they often provide either a multithreaded or an event-based programming environment [24].

Nevertheless, there is most of the time no clear separation between the operating system and the applications running on top of it. WSN operating systems generally consist of a number of selectable lightweight modules which are linked together at compilation time to create a monolithic program code which is in charge of sensing, processing, and managing communications. Some other operating systems provide an indivisible system kernel along with a set of library components for building applications [24]. More details on some common operating systems like TinyOS [33], SOS [36], Contiki [26] or LiteOS [17] can be found in [24].

14.5 Security Concerns

Until recently, most attacks on wireless sensor nodes have targeted the protocols used to communicate between nodes integrated in a network, therefore the intrinsic security of the nodes themselves is a major concern. For instance, nodes can be captured and then tampered with since they often operate in an adversary environment. We first address security of the nodes before presenting the main network-related concerns.

14.5.1 Security of Wireless Sensor Nodes

In our opinion, tamper resistance will be an important feature of sensing nodes operating in adverse environments (which is the most common case). In 2000, the authors of [18] mentioned the need for tamper resistance; however, because of the (low) cost constraint, they assumed that tamper protection for sensor nodes was limited. They even recommended not to use unattended sensors for some military missions where classified algorithms have to be used. It is true that developing tamper resistant nodes will be more expensive at the beginning of the production due to all the stages needed to be able to provide a high trust regarding the added security mechanisms. However, based on the smart card experience, one can think that the price will certainly quickly decrease when the production process will integrate these constraints.

Until 2006, node-capture has been considered for instance in [28, 53] that only the resilience of the network has been addressed. The proposed approaches were based on algorithmic solutions, e.g., through routing protocol [25], instead of a tamper-resistant mechanism, or based on redundancy to cross-check measures for consistency. The first studies [10, 11, 27] which have targeted the security of nodes themselves were done quite recently. Benenson et al. [10, 11] have developed a *"design space for physical attacks"* on nodes and have provided a framework for realistic security analysis in wireless sensor networks. Attacks (e.g., via a JTAG Test Access Port, via the Bootstrap Loader) against unattended sensor nodes in the field were discovered and countermeasures were proposed. For example, detection by its neighbors of the removal of a node through the communication protocol or by the node itself using for example an acceleration sensor, and if the removal is detected then the node is revoked of the network in the first case or it erases its confidential data in the second case. Side-channel attacks were only mentioned in the conclusions by Z. Benenson et al.; however, these threats have been considered by K. Eagles et al. [27]. In this chapter, the authors have developed a comparative threat analysis framework and a methodology to catalog threats, vulnerabilities, attacks, and countermeasures for smart cards (contact and contactless) and wireless sensor network nodes. One of the goals of their research was to determine security lessons learned from the world of smart cards that could be applied to the nodes of a wireless sensor network.

They clearly conclude that the nodes are subject to many attacks that exist on smart cards, and that tamper resistance features that are implemented within smart cards should also be considered for nodes.

Despite the very interesting conclusions of these studies, it is surprising to note that very few efforts have been made in this direction to improve the security of nodes. However, in 2010, Bialas published two papers [12, 13] in which he considered security issues of sensors used for high-risk applications. Among his contributions, he proposed Common Criteria-related security design patterns for the development of sensors' security features. This is an important step. However, applications with crucial security and reliability requirements, such as the studied sensors in charge of detecting methane in a mine [12], have not been subject to real Common Criteria evaluation and certification processes. A high level of trust for sensor nodes will only exist when Common Criteria certificates will be issued. However, the company named "Ultra Electronics 3eTI" seems close to deliver the first commercial sensing product that can reach such a level of assurance [1]. They manufacture a product called "EnergyGuard Appliance 3e-723" [2] which is a real-time energy monitoring and control system with built-in security mechanisms that allows energy managers to analyze usage at the building and base/campus level. The webpage of the product [3] claims it has been validated FIPS 140-2, Level 2, that Common Criteria EAL2 and EAL4 are pending and that it complies with DoDD 8500.1 and DoDD 8100.2.

In parallel and regardless of the conclusions of the paper by Eagles et al., three papers [30, 34, 35] have been published which exploit the network to attack the nodes themselves. Here is what the authors have been able to achieve. In [34], they compromised a Von Neumann architecture-based sensor with a classical stack attack. In [35], it is explained how a mal-packet carrying only specially crafted data can exploit memory-related vulnerabilities and utilize existing application code in a Harvard-based architecture sensor to propagate itself without disrupting sensor's functionalities. In [30], the authors succeeded in achieving a remote code injection attack on a Harvard-based architecture sensor, which was considered impossible before. This attack enables adversary to gain full control of the target sensor and for example to inject a worm that can then propagate through the wireless sensor network and possibly create a sensor botnet, or eavesdrop the network, etc.

To counter these recent attacks, Hu et al. present in [39] the design and implementation of a trusted sensor node, called trustedFleck. It uses a commodity trusted platform module (TPM) chip to extend the capabilities of a standard wireless sensor node (Fleck) to provide security services such as message integrity, confidentiality, authenticity, and system integrity. In addition, they provide services like secure software update and remote attestation.

The reader interested in software remote attestation approaches like SWATT [58] and ICE-based [57] schemes must be aware that attacks [19] exist against them and that a hardware root of trust is certainly the best reliable solution. Recently a debate [31, 52] concluded that while software-based code attestation is a useful security primitive, its design principles are not yet fully understood. The last advance [47] on this topic at the time of writing this chapter does not target sensor nodes.

14.5.2 Security in Networks of Wireless Sensor Nodes

As mentioned in the introduction of this section, most of the classical attacks that have been studied until now are related to the network protocols and not to the nodes themselves. Classical attacks like replay, packet injection, or corruption are well known and will thus not be described here. It should also be noted that operations like data aggregation and clock synchronization [44] will not be presented here in spite of the security issues that they raise, because they are application specific.

14.5.2.1 Denial-of-Service Attacks

A first class of security concerns in a sensors network is related to availability. Indeed, it is very easy for an adversary to stop the network operation with Denial-of-Service (DoS) attacks [66].

For example at physical layer, a jamming attack can be achieved by interfering on radio frequencies used by the sensor network. This kind of attack is very efficient since in most of the topologies, it does not require an important number of "attacking nodes" to perform it. However, some countermeasures to make it more complex have been developed, like:

- using spread-spectrum communication [e.g., Frequency-Hopping Spread Spectrum (FHSS)] which forces attackers to jam on a wide frequency band;
- jamming detection which enables the nodes to be switched in low consumption mode and awakened periodically to check if the jamming attack is still in progress.

Tampering with a node is also considered a DoS attack, since once captured, it is possible to destroy it or try to compromise it. Currently suggested countermeasures consist in:

- disabling the node and deleting its information when it believes that it has been compromised. It should be noted, however, that this is not very satisfactory, since the node is no longer available, which is the goal of DoS.
- camouflaging or hiding nodes in their physical environment [9], but this is not an information technology-based countermeasure!

At link-layer level, a DoS can be achieved with collision attacks which may consist in exploiting the medium access control backoff and retransmission procedures. In addition, even if collisions occur for only a few bits, the sending node should send its packet again, thus consuming more energy (this is an exhaustion attack). Fortunately, this can be partially addressed using error correcting codes and best-effort delivery protocols.

At network-layer level, a DoS can be achieved using routing loop attacks [44] to create loops in message routes so that messages are constantly forwarded around this loop, draining batteries of the nodes involved in the loop, and preventing the message from reaching its final destination.

At transport-layer level, a flooding attack can be performed by repeatedly requesting new connections to a given node which then undergoes a memory exhaustion and thus refuses further connections from legitimate nodes. A desynchronization attack can also be achieved to elicit resource-costly retransmissions [24].

14.5.2.2 Routing Attacks

There are a large number of possible attacks on routing protocols [6]. Here are a few:

- the blackhole attack in which an attacker attempts to be on the path of one or more routes to drop all the traffic so that the transmitted data never reach their destination. A variant is the selective forwarding attack, in which the attacker drops only the data matching certain criteria. This latter is more difficult to detect than a blackhole attack.
- the sinkhole attack is another variant of the blackhole attack, in which the traffic is not dropped, but disrupted or tampered.
- the rushing attack [40] in which an attacker exploits the nature of the route discovery procedure of on-demand routing protocols to increase its probability to be chosen as an intermediary node between a source and a destination.
- the sybil attack consists in an attacker presenting multiple (impersonated or false) identities to the other nodes of the network to influence the decision taken in cooperation inside the network (for instance to become a part of the route). Impersonation can be done with node or base station identities [44]. A variant of the sybil attack exists for geographic routing protocols where an attacker claims to be at several locations (instead of impersonating an identity) simultaneously with the hope to be chosen as a forwarding node.
- the wormhole attack in which two colluding attackers using an out-of-band channel between them, divert most of the traffic from the rest of the network. This then enables attacks like blackhole, sinkhole, etc.

While using wireless link encryption and authentication mechanisms can prevent most attacks from an outsider of the network, it cannot be effective against an inside attack, coming from a compromised node. As seen in this section, there are many possibilities to compromise a node and thus an intrusion detection system (IDS) is clearly an additional security mechanism that can help protecting the system.

14.5.2.3 Conclusion on Security

To the best of our knowledge, at the time of writing these lines, there is not yet any publicly available cheap wireless sensor node which would implement tamper-resistant features so as to provide a high level of trust and that could be deployed in a real network to defeat all aforementioned attacks.

Acknowledgments The authors want to thank the reviewers for their constructive comments which were helpful to improve this chapter.

References

1. 3eti: Company overview http://www.ultra-3eti.com/assets/1/7/3eTI_-_Company_Overview_(07--26-2011).pdf
2. Datasheet of energyguard appliance 3e–723 http://www.ultra-3eti.com/assets/1/7/3e-723_EnergyGuard.pdf
3. Energyguard appliance 3e–723 webpage product http://www.ultra-3eti.com/products/sensor_networks/energyguard_appliance/
4. Globalsecurity.org. sound surveillance system (sosus). http://www.globalsecurity.org/intell/systems/sosus.htm
5. Largest tiny network yet (2001). http://webs.cs.berkeley.edu/800demo/
6. Secure routing in wireless sensor networks: attacks and countermeasures (2003). doi: 10.1109/SNPA.2003.1203362. http://dx.doi.org/10.1109/SNPA.2003.1203362
7. Integrated smart sensing systems (2007). http://dpi.projectforum.com/isss/11
8. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38(4), 393–422 (2002). doi: 10.1016/S1389-1286(01)00302-4. http://www.sciencedirect.com/science/article/pii/S1389128601003024
9. Anjum, F., Sarkar, S.: Security in sensor networks. In: Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions. John Wiley & Sons.
10. Becher, E., Benenson, Z., Dornseif, M.: Tampering with motes: Real-world physical attacks on wireless sensor networks. In: in 3rd International Conference on Security in Pervasive Computing (SPC (2006).
11. Benenson, Z., Cholewinski, P.M., Freiling, F.C.: Vulnerabilities and attacks in wireless sensor networks. Wireless Sensors Networks Security pp. 22–43 (2007). http://www1.informatik.uni-erlangen.de/filepool/publications/zina/attacker-models-bookchapterIOS_Press.pdf
12. Bialas, A.: Common criteria related security design patterns alidation on the intelligent sensor example designed for mine environment. Sensors 10(5), 4456–4496 (2010). doi: 10.3390/s100504456. http://www.mdpi.com/1424-8220/10/5/4456/
13. Bialas, A.: Intelligent sensors security. Sensors 10(1), 822–859 (2010). doi: 10.3390/s100100822. http://www.mdpi.com/1424-8220/10/1/822/
14. Bisscheroux, M.: Largest deployment of myrianed wireless nodes (2010). http://wsn.chess.nl/?p=50
15. Bonnet, P., Gehrke, J.E., Seshadri, P.: Querying the physical world. IEEE Journal of Selected Areas in Communications 7(5),10–15 (2000).
16. Burrell, J., Brooke, T., Beckwith, R.: Sensor and actuator networks- Vineyard computing: sensor networks in agricultural production. IEEE Pervasive Computing 3(1), 38–45 (2004). doi: http://dx.doi.org/10.1109/MPRV.2004.1269130
17. Cao, Q., Abdelzaher, T.: liteos: a lightweight operating system for c++ software development in sensor networks. In: Proceedings of the 4th international conference on Embedded networked sensor systems, SenSys '06, pp. 361–362. ACM, New York, NY, USA (2006). doi: http://doi.acm.org/10.1145/1182807.1182855.
18. Carman, D.W., Kruus, P.S., Matt, B.J.: Constraints and approaches for distributed sensor network security. Tech. Rep. 010, NAI Labs, The Security Research Division Network Associates, Inc. (2000). http://www.cs.umbc.edu/courses/graduate/CMSC691A/Spring04/papers/nailabs_report_00-010_final.pdf
19. Castelluccia, C., Francillon, A., Perito, D., Soriente, C.: On the difficulty of software-based attestation of embedded devices. In: Proceedings of the 16th ACM conference on Computer and communications security, CCS '09, pp. 400–409. ACM, New York, NY, USA (2009). doi: http://doi.acm.org/10.1145/1653662.1653711.

20. Chandrakasan, A., Amirtharajah, R., Cho, S., Goodman, J., Konduri, G., Kulik, J., Rabiner, W., Wang, A.: Design considerations for distributed micro-sensor systems. In: Proceedings of the IEEE 1999 Custom Integrated Circuits Conference, pp. 279-286 (1999).
21. Chatzigiannakis, I., Fischer, S., Koninis, C., Mylonas, G., Pfisterer, D.: Wisebed: An open large-scale wireless sensor network testbed (2010). http://dx.doi.org/10.1007/978-3-642-11870-8_6.10.1007/978-3-642-11870-8_6
22. Clark, C.W., Mellinger, D.K.: Application of navy iuss for whale research. The Journal of the Acoustical Society of America 96(5), 3315–3315 (1994). doi: 10.1121/1.410808. http://link.aip.org/link/?JAS/96/3315/1
23. Crow, B., Widjaja, I., Kim, J., Sakai, P.: IEEE 802.11 Wireless Local Area Networks. IEEE Communications Magazine pp. 116–126 (1997).
24. Dargie, W., Poellabauer, C.: Fundamentals of Wireless Sensor Networks: Theory and Practice. Wireless Communications and Mobile Computing. Wiley (2010). http://books.google.fr/books?id=8c6k0EVr6rMC
25. Deng, J., Han, R., Mishra, S.: A performance evaluation of intrusion-tolerant routing in wireless sensor networks. In: Proceedings of the 2nd international conference on Information processing in sensor networks, IPSN'03, pp. 349–364. Springer-Verlag, Berlin, Heidelberg (2003). http://dl.acm.org/citation.cfm?id=1765991.1766015
26. Dunkels, A., Gronvall, B., Voigt, T.: Contiki - a lightweight and flexible operating system for tiny networked sensors. In: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, LCN '04, pp. 455–462. IEEE Computer Society, Washington, DC, USA (2004). doi: http://dx.doi.org/10.1109/LCN.2004.38.
27. Eagles, K., Markantonakis, K., Mayes, K.: A comparative analysis of common threats, vulnerabilities, attacks and countermeasures within smart card and wireless sensor network node technologies. In: Proceedings of the 1st IFIP TC6 /WG8.8 /WG11.2 international conference on Information security theory and practices: smart cards, mobile and ubiquitous computing systems, WISTP'07, pp. 161–174. Springer-Verlag, Berlin, Heidelberg (2007). http://dl.acm.org/citation.cfm?id=1763190.1763209
28. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: In Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 41–47. ACM Press (2002).
29. Farahani, S.: ZigBee Wireless Networks and Transceivers. Newnes, Newton, MA, USA (2008).
30. Francillon, A., Castelluccia, C.: Code injection attacks on harvard-architecture devices. In: Proceedings of the 15th ACM conference on Computer and communications security, CCS '08, pp. 15–26. ACM, New York, NY, USA (2008). doi: http://doi.acm.org/10.1145/1455770.1455775. http://doi.acm.org/10.1145/1455770.1455775
31. Francillon, A., Castelluccia, C., Perito, D., Soriente, C.: Comments on efutation of on the difficulty of software-based attestation of embedded devices (2010).
32. Frank, R.: Understanding Smart Sensors. Measurement Science and Technology 11(12), 1830 (2000). doi: http://dx.doi.org/10.1088/0957-0233/11/12/711
33. Gay, D., Levis, P., Culler, D.: Software design patterns for tinyos. ACM Trans. Embed. Comput. Syst. 6 (2007). doi: http://doi.acm.org/10.1145/1274858.1274860.
34. Goodspeed, T.: Exploiting wireless sensor networks over 802.15.4. In: ToorCon 9 (2007).
35. Gu, Q., Noorani, R.: Towards self-propagate mal-packets in sensor networks. In: Proceedings of the first ACM conference on Wireless network security, WiSec '08, pp. 172–182. ACM, New York, NY, USA (2008). doi: http://doi.acm.org/10.1145/1352533.1352563. http://doi.acm.org/10.1145/1352533.1352563
36. Han, C.C., Kumar, R., Shea, R., Kohler, E., Srivastava, M.: A dynamic operating system for sensor nodes. In: Proceedings of the 3rd international conference on Mobile systems, applications, and services, MobiSys '05, pp. 163–176. ACM, New York, NY, USA (2005). doi: http://doi.acm.org/10.1145/1067170.1067188.
37. Hefeeda, M., Bagheri, M.: Wireless sensor networks for early detection of forest fires. In: IEEE 4th International Conference on Mobile Adhoc and Sensor Systems, MASS 2007, 8–11 October 2007, Pisa, Italy, pp. 1–6. IEEE (2007). doi: http://dx.doi.org/10.1109/MOBHOC.2007.4428702

38. Heukoop, C.V.: Alwen 1000 node experiment. In: Elektronica (2010). http://wsn.chess.nl/wp-content/uploads/2010/02/AlwEN-1000node-exp-Elektronica-janfeb2010.pdf
39. Hu, W., Tan, H., Corke, P., Shih, W.C., Jha, S.: Toward trusted wireless sensor networks. ACM Trans. Sen. Netw. 7, 5:1–5:25 (2010). doi: http://doi.acm.org/10.1145/1806895.1806900.
40. Hu, Y.C., Perrig, A., Johnson, D.B.: Rushing attacks and defense in wireless ad hoc network routing protocols. In: Proceedings of the 2nd ACM workshop on Wireless security, WiSe '03, pp. 30–40. ACM, New York, NY, USA (2003). doi: http://doi.acm.org/10.1145/941311.941317. http://doi.acm.org/10.1145/941311.941317
41. Johnson, P., Andrews, D.C.: Remote continuous physiological monitoring in the home. Journal of Telemedicine and Telecare 2(2), 107–113 (1996).
42. Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L.S., Rubenstein, D.: Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. SIGPLAN Not. 37, 96–107 (2002). doi: http://doi.acm.org/10.1145/605432.605408. http://doi.acm.org/10.1145/605432.605408
43. Kumar, S., Shepherd, D.: SensIT: Sensor Information Technology for the warfighter. In: Proceedings of the 4th Conference on Information Fusion, pp. 3–9. Montreal, Canada (2001).
44. Larsson, A.: Report on the state of the art of security in sensor, networks (2011).
45. Lee, K.: Ieee 1451: A standard in support of smart transducer networking. In: Proceedings of IEEE Instrumentation and Measurement, vol. 2, pp. 525–528. IEEE (2000). http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=848791
46. Lennvall, T., Svensson, S., Hekland, F.: A comparison of WirelessHART and ZigBee for industrial applications. In: Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on, pp. 85–88 (2008). doi: http://dx.doi.org/10.1109/WFCS.2008.4638746
47. Li, Y., McCune, J.M., Perrig, A.: Viper: verifying the integrity of peripherals' firmware. In: Proceedings of the 18th ACM conference on Computer and communications security, CCS '11, pp. 3–16. ACM, New York, NY, USA (2011). doi: http://doi.acm.org/10.1145/2046707.2046711
48. Lopez, J., Roman, R., Alcaraz, C.: Analysis of security threats, requirements, technologies and standards in wireless sensor networks. In: A. Aldini, G. Barthe, R. Gorrieri (eds.) Foundations of Security Analysis and Design V, Lecture Notes in Computer Science, vol. 5705, pp. 289–338. Springer Berlin/Heidelberg (2009). http://dx.doi.org/10.1007/978-3-642-03829-7_10.
49. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., Anderson, J.: Wireless sensor networks for habitat monitoring. In: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, WSNA '02, pp. 88–97. ACM, New York, NY, USA (2002). doi: http://doi.acm.org/10.1145/570738.570751.
50. Mills, K.: A brief survey of self-organization in wireless sensor networks. Wireless Communications and Mobile Computing 7(7), 823–834 (2007).
51. Nishimura, C.E., Conlon, D.M.: Iuss dual use: Monitoring whales and earthquakes using sosus. Marine Technology Society Journal 27(4), 13–21 (1994).
52. Perrig, A., van Doorn, L.: Refutation of n the Difficulty of Software-Based Attestation of Embedded Devices (2010). http://sparrow.ece.cmu.edu/group/pub/perrig-vandoorn-refutation.pdf
53. Perrig, A., Stankovic, J., Wagner, D.: Security in wireless sensor networks. COMMUNICATIONS OF THE ACM 47(6), 53–57 (2004).
54. Rabaey, J.M., Ammer, M.J., da Silva, J.L., Patel, D., Roundy, S.: Picoradio supports ad hoc ultra-low power wireless networking. Computer 33, 42–48 (2000). doi: 10.1109/2.869369. http://dl.acm.org/citation.cfm?id=619053.621512
55. Roundy, S., Steingart, D., Frechette, L., Wright, P., Rabaey, J.: Power Sources for Wireless Sensor Networks. pp. 1–17 (2004). http://www.springerlink.com/content/b0utgm8ahnphll3l
56. Roundy, S., Wright, P.K., Rabaey, J.: A study of low level vibrations as a power source for wireless sensor nodes. Computer Communications 26(11), 1131–1144 (2003). doi: http://www.sciencedirect.com/science/article/pii/S0140366402002487
57. Seshadri, A., Luk, M., Perrig, A., Doorn, L., Khosla, P.: Scuba: Secure code update by attestation in sensor networks. In: in Proceedings of ACM Workshop on Wireless Security (WiSe6). ACM, pp. 85–94. Press (2006).

58. Seshadri, A., Perrig, A., Doorn, L.V., Khosla, P.: Swatt: Software-based attestation for embedded devices. In: In Proceedings of the IEEE Symposium on Security and Privacy (2004).
59. Shih, E., Bahl, P., Sinclair, M.J.: Wake on wireless: an event driven energy saving strategy for battery operated devices. In: MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking, pp. 160–171. ACM, New York, NY, USA (2002). doi: http://dx.doi.org/10.1145/570645.570666
60. Shih, E., Cho, S.H., Ickes, N., Min, R., Sinha, A., Wang, A., Chandrakasan, A.: Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In: Proceedings of the 7th annual international conference on Mobile computing and networking, MobiCom '01, pp. 272–287. ACM, New York, NY, USA (2001). doi: http://doi.acm.org/10.1145/381677.381703.
61. Simon, G., Maróti, M., Lédeczi, A., Balogh, G., Kusy, B., Nádas, A., Pap, G., Sallai, J., Frampton, K.: Sensor network-based countersniper system. In: Proceedings of the 2nd international conference on Embedded networked sensor systems, SenSys '04, pp. 1–12. ACM, New York, NY, USA (2004). doi: http://doi.acm.org/10.1145/1031495.1031497.
62. Song, J., Han, S., Mok, A., Chen, D., Lucas, M., Nixon, M., Pratt, W.: Wirelesshart: Applying wireless technology in real-time industrial process control. In: Proceedings of the 2008 IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 377–386. IEEE Computer Society, Washington, DC, USA (2008). doi: 10.1109/RTAS.2008.15. http://dl.acm.org/citation.cfm?id=1440456.1440604
63. Song, J., Han, S., Mok, A.K., Chen, D., Lucas, M., Nixon, M.: WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control. In: Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS '08. IEEE, vol. 0, pp. 377–386. IEEE, Los Alamitos, CA, USA (2008). doi: http://dx.doi.org/10.1109/RTAS.2008.15
64. Surhone, L., Tennoe, M., Henssonow, S.: Isa100.11a. VDM Verlag Dr. Mueller AG & Co. Kg (2010). http://books.google.fr/books?id=F_BMYgEACAAJ
65. Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J., Welsh, M.: Monitoring volcanic eruptions with a wireless sensor network. In: Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on, pp. 108–120. IEEE (2005). doi: http://dx.doi.org/10.1109/EWSN.2005.1462003
66. Wood, A.D., Stankovic, J.A.: Denial of service in sensor networks. Computer 35, 54–62 (2002). doi: http://dx.doi.org/10.1109/MC.2002.1039518.
67. Wright, R., Flynn, L., Garbeil, H., Harris, A., Pilger, E.: Automated volcanic eruption detection using MODIS. Remote Sensing of Environment 82(1), 135–155 (2002). doi: http://dx.doi.org/10.1016/S0034-4257(02)00030-5
68. Ye, W., Heidemann, J., Estrin, D.: Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans. Netw. 12, 493–506 (2004). doi: http://dx.doi.org/10.1109/TNET.2004.828953.