

Article

Critical Infrastructure Surveillance Using Secure Wireless Sensor Networks

Michael Niedermeier ^{1,*}, Xiaobing He ¹, Hermann de Meer ¹, Carsten Buschmann ²,
Klaus Hartmann ³, Benjamin Langmann ³, Michael Koch ⁴, Stefan Fischer ⁵
and Dennis Pfisterer ⁵

¹ Department of Computer Networks and Computer Communications, University of Passau,
Innstr. 43, 94032 Passau, Bayern, Germany; E-Mails: Xiaobing.He@uni-passau.de (X.H.);
demeer@uni-passau.de (H.M.)

² Coalesenses GmbH, Maria-Goeppert-Str. 1, 23562 Lübeck, Germany;
E-Mail: buschmann@coalesenses.com

³ Center for Sensor Systems, University of Siegen, 57076 Siegen, Germany;
E-Mails: hartmann@zess.uni-siegen.de (K.H.); langmann@zess.uni-siegen.de (B.L.)

⁴ SINUS Messtechnik GmbH, Föpplstr. 13, 04347 Leipzig, Germany
E-Mail: michael.koch@sinusmess.de

⁵ Institute of Telematics, University of Lübeck, 23562 Lübeck, Germany;
E-Mails: fischer@itm.uni-luebeck.de (S.F.); pfisterer@itm.uni-luebeck.de (D.P.)

* Author to whom correspondence should be addressed; E-Mail: michael.niedermeier@uni-passau.de;
Tel.: +49-851-509-3056.

Academic Editors: Zbigniew Kotulski, Bogdan Ksiezopolski and Pascal Lafourcade

Received: 8 June 2015 / Accepted: 9 November 2015 / Published: 25 November 2015

Abstract: In this work, a secure wireless sensor network (WSN) for the surveillance, monitoring and protection of critical infrastructures was developed. To guarantee the security of the system, the main focus was the implementation of a unique security concept, which includes both security on the communication level, as well as mechanisms that ensure the functional safety during its operation. While there are many theoretical approaches in various subdomains of WSNs—like network structures, communication protocols and security concepts—the construction, implementation and real-life application of these devices is still rare. This work deals with these aforementioned aspects, including all phases from concept-generation to operation of a secure wireless sensor network. While the key focus of this paper lies on the security and safety features of the WSN, the detection, localization and

classification capabilities resulting from the interaction of the nodes' different sensor types are also described.

Keywords: wireless sensor networks; security; functional safety; networking

1. Introduction

Wireless sensor networks are networking structures comprised of many small and low-cost sensor nodes, which have limited computational power and energy supply. The main goal of these networks is sensing, actuating and sending of environmental information to a data sink, which then processes it. During the last years, the application areas have evolved from military surveillance to environmental and animal monitoring. While these networks have a high potential—both from a research and economic perspective—they are currently rarely deployed, especially in high-security applications. The reason behind this is that the biggest advantage of WSNs—their autonomous and unattended operation—in turn also opens up many attack possibilities, like, e.g., tampering, physical manipulation and node compromise, due to the unavoidable disappearance of a security perimeter.

To prove that WSNs are applicable, even in high-security areas, this paper describes the comprehensive development of a secure WSN, including hardware design, software implementation as well as practical tests in laboratory and field environments. To cope with the aforementioned security issues, the system not only implements features from the field of communication security, but also includes functional safety and self-protection functions. All of these are fused in the system, creating a comprehensive security concept, which guarantees confidentiality, integrity, and availability of the WSN at any time. In addition, the reporting and logging mechanisms allow for users to constantly monitor the activities inside the sensor network, both in online and offline. Moreover, the usability of the system is not impaired by the inherent complexity of the WSN, as it can be configured and operated from a central command center. The combination of these features creates a uniquely secure, flexible and usable system.

The remainder of this paper is structured in a way that describes all stages of the system realization: Section 2 explains the requirements of our system. Section 3 covers all aspects of the system development and Section 4 demonstrates the abilities of the utilized sensors. Section 5 shows both the laboratory assessments as well as the field trials of the system. Section 6 presents results of the aforementioned tests. The final two sections, Sections 7 and 8 present related work to protect critical infrastructures and a conclusion to the paper.

2. Requirement Analysis

As stated in Section 1, the application areas of WSNs have broadened from solely scientific tasks to non-critical, yet productive applications and are currently on the verge to be considered as solution to achieve critical objective, such as the protection of critical infrastructures, too. In this paper, a WSN is presented that fulfills the requirements required for the latter purpose. The WSN is capable to protect

both the borders of an arbitrary object (against intruders) as well as itself (against manipulations). While the WSN can be used to surveil any kind of bordered area, this combination of features ideally fits its usage in critical infrastructures protection.

While this work's general focus are the security and safety features, the main functional goals of the system are defined by three tasks of the WSN: (i) detection; (ii) localization; and (iii) classification of objects in a predefined area. The system must be able to detect and locate persons or cars with certain accuracy. In our case, all values for deviations or timings are chosen in a way that the goal on the user's side—which is to be able to organize a well-directed response to a penetration of the surveillance area—is possible. For the given case, a deviation of ≤ 5 m is therefore considered acceptable.

In addition, it is required to calculate the trajectory of the trespassing object to indicate its direction. In order to provide not only accurate but also timely data, the system is required to fulfill certain real-time criteria. The delay between object-entry and the signaling of the event to the user was fixed at a maximum delay of 3 s. Due to the special hardware used in WSNs, several additional system requirements have to be fulfilled. Among them is primarily a reliable, scalable and efficient communication architecture ensuring that all events occurring inside the monitored area are not only registered, but also that the messages of these events are transferred reliably to the base station [1].

The security of the system is of major importance, especially when the unattended operation of the system is taken into account. The system's overall security concept, which is described in Section 3.3.2, therefore comprises not only IT security measures but also functional safety aspects. This is required to protect the system from coincidental or targeted physical damage and hardware failures. Additionally, because of the hardware of the sensor nodes, which only relies on limited battery supply, energy efficiency has to be carefully considered in both the hardware and software design [2].

The following Section 3 shows how the aforementioned requirements are fulfilled and implemented in a prototype system.

3. Network Design and Architecture

In this section, the development of the wireless sensor network is described in detail. This includes an overview of the overall concept of the sensor network, a fine-grained presentation of the hard- and software concepts and the proposed secure detection, localization and classification algorithms.

3.1. Network Overview

This section presents the general structure of the system to give an overview of its functionalities. As depicted in Figure 1, the system is based on a hierarchical structure, that consists of a single command center c , several clusterheads ch_i on the intermediate level and for each clusterhead ch_i multiple sensor nodes $n_{i,j}$ on the bottom level. In the prototype system, whose realization is described later on, a total number of 100 nodes is used (10 clusterheads ch_i , $1 \leq i \leq 10$ with each 10 sensor nodes $n_{i,j}$, $1 \leq i \leq 10$, $1 \leq j \leq 10$). Due to the system's scalability, the system can however also be employed using a much larger number of nodes.

The functions of the system—detection, localization and classification of trespassers—are done centrally at the command center c , using data provided by the sensor nodes $n_{i,j}$. The clusterheads ch_i

fulfill managing and information-aggregating functions in the data delivery process from n_{ij} to c . The details concerning the hard- and software realization is covered in the following Sections 3.2 and 3.3.

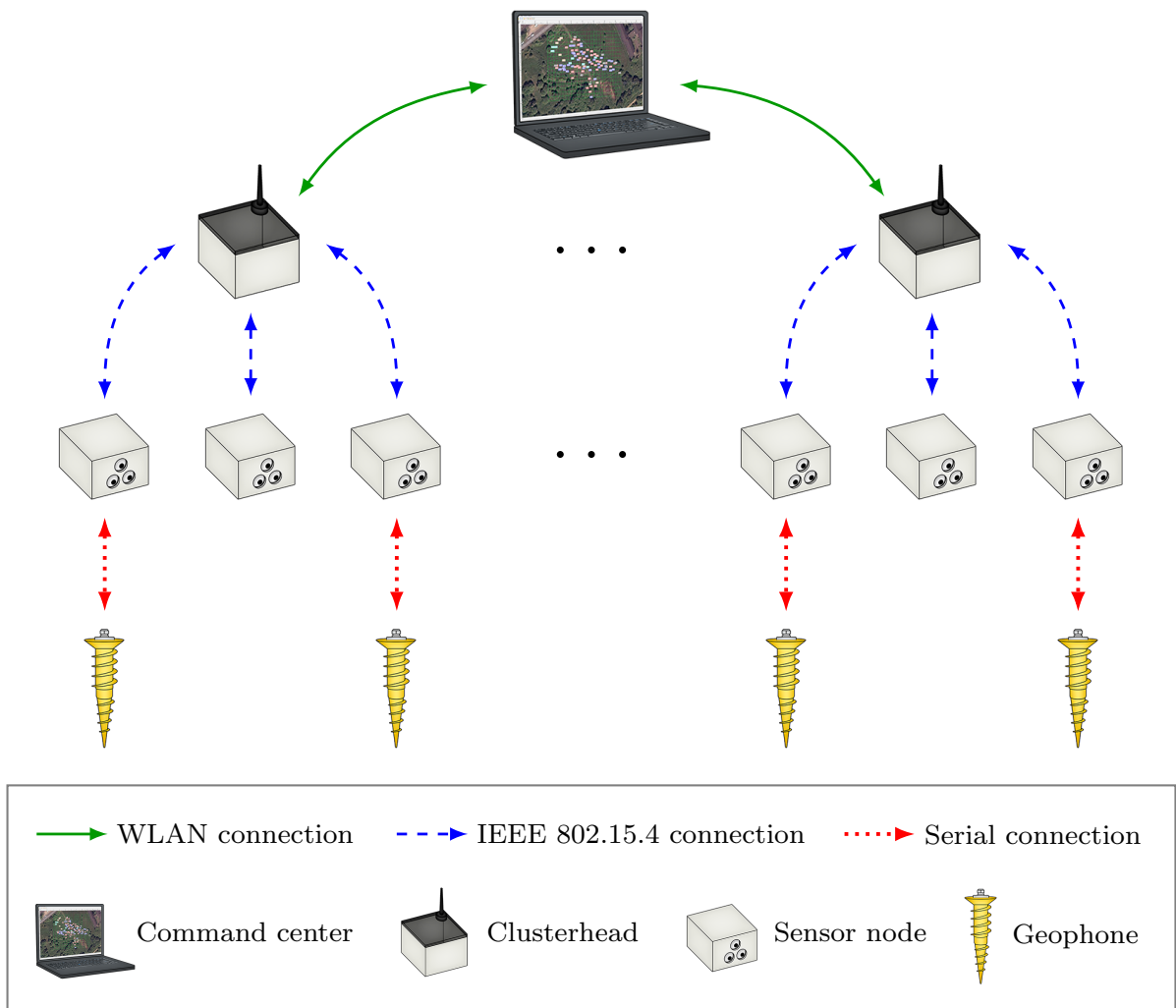


Figure 1. Overall system structure.

The sensor nodes are equipped with different sensor combinations as shown in Table 1. A more detailed description of the sensors is presented in Section 4.

Table 1. Used sensor node types.

Sensor Type	Number of Nodes Equipped	Node Type Equipped
AMR	90	Sensor nodes
Accelerometer	90	Sensor nodes
Single-PIR	45	Sensor nodes
Multi-PIR	45	Sensor nodes
Longrange-PIR	10	Sensor nodes
Geophone	50	Sensor nodes
GPS	10	Clusterheads

3.2. Hardware

In this section, the hardware concept is described in the order of the previously shown three levels: command center, clusterheads, and sensornodes.

3.2.1. Command Center

The command center c aggregates data from the clusterheads ch_i and displays the information to the user (cf. Figure 2). This information is presented using the detection algorithms described in Section 3.4 and delivered to a network visualization software called *Spyglass* [3]. The existing open-source Spyglass was extended to have a control window that allows it to send commands to the network, like, e.g., restarting nodes, sending messages to a number of nodes, or re-programming nodes.



Figure 2. The visualization program Spyglass.

3.2.2. Clusterheads

Each clusterhead ch_i consists of two components: an embedded-PC board (ARM-based CPU with 800 MHz, 512 MB RAM, 512 MB flash memory, power requirement of about 1.5 W) and an iSense sensor node. The embedded PC maintains the WiFi connection to the command center c , preprocesses sensor data, and performs some network management tasks. To communicate with the sensor nodes, an iSense sensor network device is attached to the CPU board. It consists of an iSense Core Module (CM30I, [4]), an iSense Gateway Module (GM20-P, [5]) for USB connection to the embedded PC,

and an iSense GPS Module (GPSM10S, [6]) for time synchronization purposes. All aforementioned components are depicted in Figure 3.

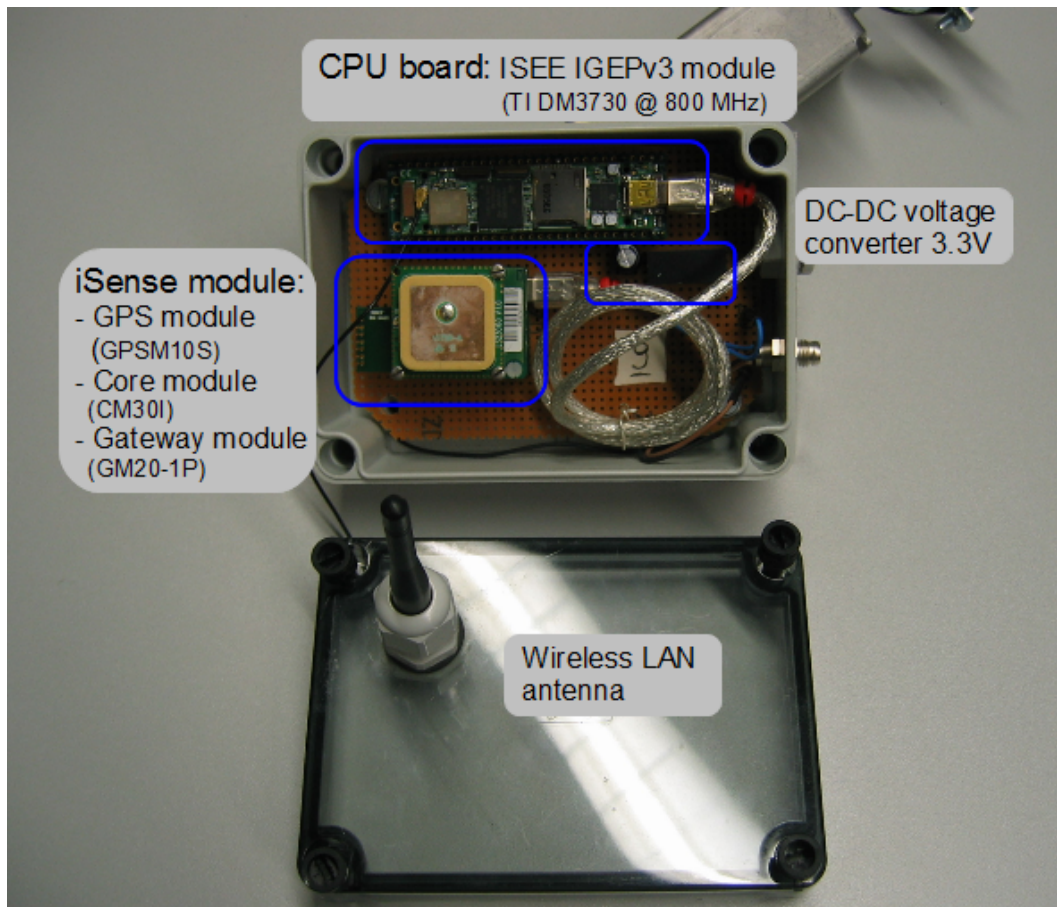


Figure 3. Hardware components of a clusterhead.

3.2.3. Sensor Nodes

Several different configurations of sensor nodes have been designed, which differ in the type of attached sensors and housing (Figure 4). The main type of housing (Fibox PC121210) is used for the majority of sensor nodes. These consist of 3 or 4 different hardware modules of the iSense modular sensor network hardware platform [7].

The Core Module includes a Jennic JN5148 32-bit RISC controller [8] operating at a frequency of 16 MHz and provides 192 kB of ROM, 512 kB of flash memory, 128 kB of RAM for instructions and data, and an IEEE 802.15.4 [9] compliant radio interface. The radio chip operates at a frequency of 2.4 GHz, offers 16 different radio channels, provides a data transfer rate of 250 kB/s and includes a hardware AES [10] engine. Besides those, the Core Module contains an ultra-stable real-time clock (RTC) (typical 6 ppm), a voltage regulator and a 34-pin connector on each side to connect other modules. The second type of housing contains long-range Siemens IS392 [11] PIR sensors that are attached to the iSense Sensor and Core Module. In addition, an accelerometer is included in the housing.

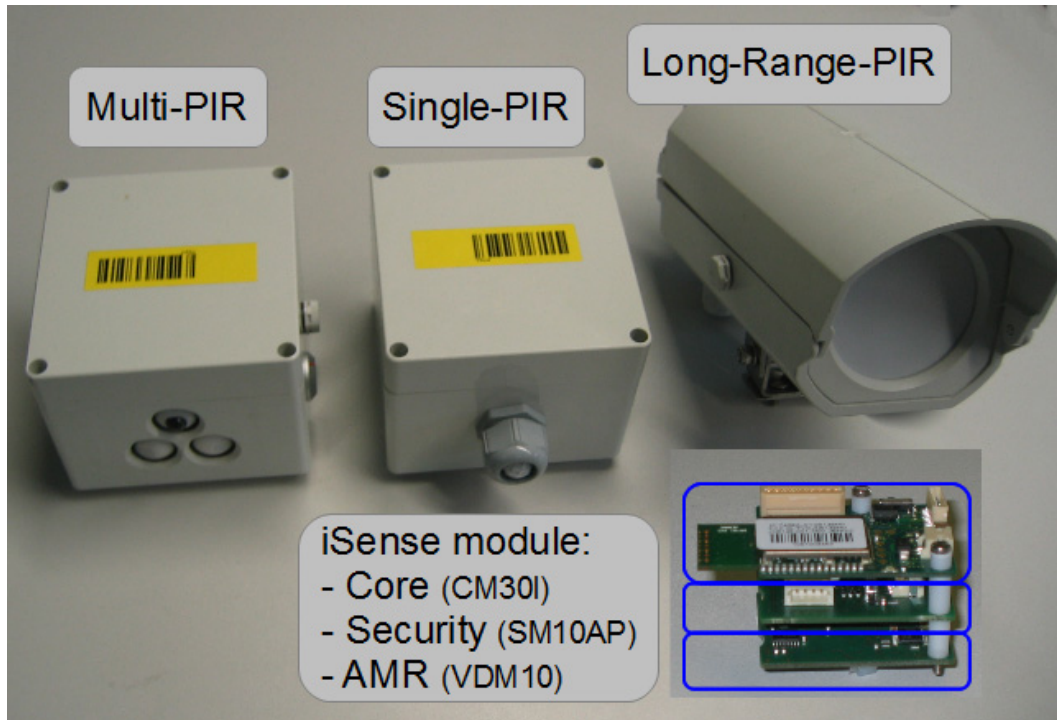


Figure 4. Overview of the different sensor nodes and iSense module components.

3.2.4. Sensors

In the sensor network, the previously described four types of sensors are utilized for various purposes. The general detection of activity in the observed area is performed by PIR sensors, which can be differentiated into three types: Single-PIRs (Panasonic AMN34111, [12]) with a range up to 10 m, Multi-PIRs (AMN31111, [12] + 2 × AMN33111, [12]) with a range of approximately 5 m and Long-Range-PIRs (Siemens IS392) with a range up to 50 m. With the Multi-PIRs, the directions of moving objects can be estimated and Long-Range-PIRs play a significant role in the detection of new objects entering the surveillance area. Figure 11 shows detection results for these different PIR sensors.

The second type of sensors used for detection purposes are geophones. They are comprised of three seismic capsules (SM-24, Sensor Nederland, [13]) in an orthogonal arrangement, an analog signal conditioning part and a microcontroller-based (ARM-Cortex M3 MCU) signal processing unit. The complete geophone is housed in a protection enclosure and is connected to the sensor node via a communication cable that also provides the required power. For a sufficient coupling to the ground, a soil drilling tool in combination with a screw-shaped housing is used.

The Vehicle Detection Modules [14] are used for object classification, since only moving metallic objects influence their measurements. They are based on a 2-axis Phillips KMZ52 anisotropic magneto-resistive sensor bridge that is combined with two amplifier stages as well as circuitry for de-gaussing and earth-magnetic field compensation. It exploits the fact that large ferro-magnetic objects distort the earth-magnetic field to detect such objects by observing field changes. To achieve a detection range of more than 5 m, it amplifies the bridge output by a factor of approximately 40,000 and features a sensitivity of 786.2 mV/(kA/m) at a bandwidth of 1 kHz. Because the module typically has a current consumption of 20 – 25 mA during operation, it is activated only on PIR sensor events within less than

180 ms. Figures 14 and 15 show the typical output of the module when a car passes by in a distance of 4 m. The vehicle detection itself is done in software on the sensor node's controller by using two ADC inputs to sample the module's output on both channels.

The last sensor is an accelerometer, which is part of every sensor node. Its sole purpose is the detection of physical tampering with the sensor nodes, since even small movements of the device trigger the accelerometer.

3.3. Software

In this section, we describe the software architecture of the sensor network, which consists of three parts:

- The hierarchical communication structure.
- The security concept for providing communication security and functional safety (guaranteeing the detection of malfunctions in the sensor or node hardware).
- The detection algorithm. This uses data gathered by the sensor nodes and relates between the sensor clusters to derive a trespasser's position and current direction of movement.

3.3.1. Hierarchical Communication Structure

As discussed earlier, the hierarchical communication structure ensures network scalability and is comprised of the command center, clusterheads (embedded PC and sensor node), and the sensor nodes which are associated with the clusterheads. The communication between command center and clusterheads is realized via standard TCP/IP-based connections over WiFi while the communication in the sensor network is based on the energy-efficient IEEE 802.15.4 standard. This structure is also depicted in Figure 1: sensor readings from the sensor nodes are forwarded to the sensor node attached to the clusterhead, are pre-processed, and finally forwarded to the command center, where further data aggregation and preprocessing occurs before data is visualized and presented to the operator. Vice versa, there is a data flow from the command center downstream to send configuration instructions, to reprogram the sensor node parts that are attached to the clusterhead PCs, and to wirelessly reprogram the sensor nodes.

In order not to build a proprietary system, we re-use existing open-source software and implement our additional functionality as plug-ins. The software Testbed Runtime (<https://github.com/itm/testbed-runtime>) is a result of the EU-project WISEBED [15] and allows to operate wireless sensor network testbeds. Its architecture is depicted in Figure 5.

It consists of a number of sensor nodes attached to gateways (e.g., via USB). The gateways communicate over TCP/IP with a Testbed Server that exposes the functionality of the network (e.g., sending messages to attached sensor nodes or reprogramming them, receiving messages from nodes emitted via USB, *etc.*) via SOAP-based web services. In addition, it supports so-called filters through which all data is passed that is exchanged between (i) the sensor nodes and gateway and (ii) the Testbed Server and the web service clients. A filter may alter, drop, or fabricate messages (e.g., for data aggregation).

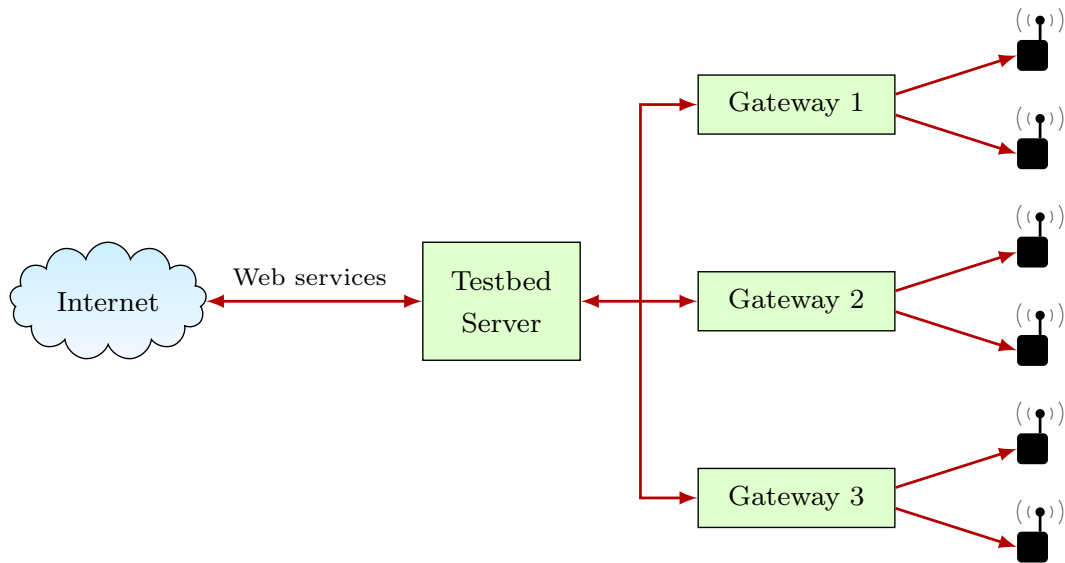


Figure 5. Architecture of the Testbed Runtime software.

This structure blends well with the overall communication infrastructure (*cf.* Figure 1). To implement our hierarchical communication structure, we use the Testbed Runtime software as the basis and implement our data aggregation and trespasser detection schemes as filters. To control the network, a client for the SOAP-based web services is required and we add this feature to the visualization framework *Spyglass*, which acts as the control center. In order to send messages to the sensor network, data is sent to the gateway node, which receives this information from Testbed Runtime via USB. For the wireless sensor network, we design a protocol on the USB connection so that data can be forwarded securely to/from the wireless sensor nodes. To initialize the communication between the clusterhead and sensor nodes, firstly, the clusterhead broadcasts a heartbeat solicitation (HBS) message to all sensor nodes. After checking the authentication and validity of the clusterhead, the sensor node replies with a heartbeat advertisement (HBA) message (with 1 status Byte and 1 signal quality Byte to indicate the status of the sensor node and the signal quality of the received HBS, respectively), as well as a clusterhead solicitation (CHS) message. The clusterhead will also check the authentication and validity of a sensor node and then sends a cluster advertisement (CHA) and a timestamp message to the sensor node. After that, the communication for the wireless sensor network is established and of course, all the messages exchanged for communication initialization are secured with the strategies described in Section 3.3.2.

3.3.2. Security

To fulfill high-security requirements (e.g., protecting critical infrastructures) IT technologies require special communication security features to guarantee the necessary protection level during the system's operation. However, securing the communication alone is not sufficient if the devices should also be transportable, usable in infrastructure-less areas, and shall be suited for unattended operation. As previously described, the combination of these requirements requires novel security concepts, which are explained in the following.

Communication Security

While the IEEE 802.15.4 standard already offers several security features, these were not used to secure the data exchange for two reasons. First, the security functions should be kept separate from the underlying communication protocol to offer flexibility. Second, the security features offered by 802.15.4 are not sufficient for the critical application scenario of the system. The system implements additional security functions to detect integrity breaches, e.g., replay attacks as well as arbitrary and targeted jamming. The communication security itself is structured in two security layers, which bundle certain security functions. The communication stack of the system is made up of four layers, from top to bottom: application-layer, security-layer, AFH-layer (Adaptive Frequency Hopping) and IEEE 802.15.4-layer. This structure guarantees a mutual independence of the sensor data gathering and evaluation, the AFH scheme and the other security features. the AFH-layer implements the adaptive frequency hopping, time synchronization, management of the currently allowed communication channels, and the procedure for the initial joining of nodes to clusters. The security layer is described next.

Security-Layer: The security-layer implements all communication security features except for the AFH mechanism, which can be easily found in [16], and we are not going to repeat it here but how AFH works in this system is tested in Section 6. The goal of the security-layer is to achieve confidentiality, integrity, and availability. To do so, a combination of several security measures is used.

To prevent messages from being forged or manipulated, the message payload is encrypted using the AES-CBC-128 cipher, which is supported by a crypto co-processor available in all sensor nodes, which makes it very fast and energy-efficient. In addition, a 4 Byte timestamp, a 16 Byte CBC-MAC AES, and a 1 Byte sequence number is added to the payload. These are used to achieve resistance against, e.g., replay attacks, delayed message sending, or manipulation of the message order.

All security measures realized by the security layer are depicted in Figure 6. The cryptographic secret key k_e is also a part of the security layer and it is used with the CCM* cipher algorithm [17]. To prevent the system from being compromised by brute-force attacks, k_e is altered before each communication round by re-encrypting the currently used key: $k_{e_{t_{n+1}}} = AES(k_{e_{t_n}})$.

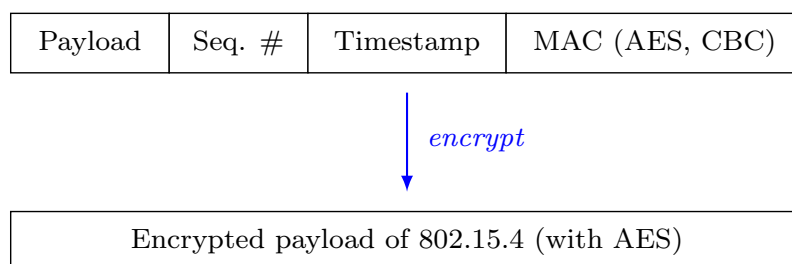


Figure 6. Communication security measures.

The communication between the clusterheads and the command center c is not realized with IEEE 802.15.4 but with standard IEEE 802.11g WiFi offering WPA2 security. As WPA2 offers all necessary features for a high protection level of the communication [18], no further security measures are applied here.

Functional Safety

The integrity features of the nodes and their respective sensors are key components of the functional safety of the wireless sensor network. The constant monitoring of the components' integrity allows to detect both random, transient as well as static errors and malfunctions in the components. The error detection then allows to control the error and thereby achieve a safe state in the sensor network. The reliability of the system is highly important, as undetected sensor or system malfunctions could easily lead to detection holes in the surveillance area. Because of that, numerous functional safety features are implemented, to uncover any system failures and induce accordant countermeasures to guarantee the proper operation of the system.

RAM: To exclude hardware-related failures due to RAM malfunctions, an integrity check for the hardware is implemented that checks the correct functionality of the RAM each time a node is started or restarted. This test is implemented using the "Checkerboard" scheme, that is outlined in Code 1. The test writes hexadecimal 5 s (0×55) in all even and hexadecimal As ($0 \times AA$) in all odd memory locations in a checkerboard pattern. After the specified parts of the RAM have been written and an additional wait time of ≥ 10 ms (which are needed to make sure that the remaining capacitance is gone before the reading starts) the patterns are read and compared. That comparison is done by *XOR*ing the i^{th} word ($0 \leq i < \lceil \frac{n}{2} \rceil$, n = number of words in memory) in the memory with the $(n - i)^{\text{th}}$ word. The expected result of this operation is identical for all words in the memory: A word consisting of only $0 \times FF$. If the result differs from the expected value, an error is assumed. After this, the test is repeated using complementary patterns (0×55 in odd locations, $0 \times AA$ in even locations).

If the RAM test fails, the node is restarted to avoid the spreading of possibly falsified information in the network. After the restart, the hardware testing starts again, until the RAM is working correctly again. If the error is permanent, the node tries to restart until its energy supply is drained.

Program data: To protect the integrity of the application running on the nodes, a checksum mechanism was implemented that checks the code inside the RAM and compares it to the original image's code stored inside the flash memory. The code check uses Fletcher's checksum that is both effective and computationally efficient, which is important due to the energy restrictions of the sensor nodes. The quality of this checksum algorithm (diagnostic coverage value) is similar to that of CRC (Cyclic Redundancy Check) [19]. The pseudo code in Code 2 shows the basic scheme of Fletcher's checksum. This check ensures that the code is not altered on its way from the flash memory to the RAM.

CPU: As the key component of every sensor node, the CPU is included in the functional safety tests. For a complete verification of the CPU, a very complex, and thereby energy- and time-consuming, procedure would be required that exhaustively tests every CPU instruction. Because of that, an extensive assessment is not implemented. Instead, a correctly working CPU is assumed, if the self-tests of both the RAM and the flash memory are finished without errors and the self-test algorithm is completed and exits as expected. To check that, the error flag is set before the test routine begins and is only reset if the program finishes at the expected exit point. If this is not the case, a CPU error is indicated, leading to a node reset, as previously explained in the RAM test procedure.

Code 1 Pseudocode of Checkerboard RAM test

```

//Checkerboard RAM test
integer i, j, k, n

while i is odd and j is even do
    write 0x55 in cell[i]
    write 0xAA in cell[j]
end while

pause(10 ms)

//Check if sum of even and odd locations is 0xFF
for k = 0 to  $\lceil \frac{n}{2} \rceil - 1$  do
    sum[k] = cell[k] + cell[n - k]
    if not_equal(sum[k], 0xFF) then
        FAILURE
    else
        //Complement cells if their sum is 0xFF
        complement(cell[k])
        complement(cell[n - k])
    end if
end for

pause(10 ms)

//Check complemented cells
for k = 0 to  $\lceil \frac{n}{2} \rceil - 1$  do
    sum[k] = cell[k] + cell[n - k]
    if not_equal(sum[k], 0xFF) then
        FAILURE
    end if
end for

PASS

```

Sensors: The most important part of the integrity checking process is the testing of the sensors to prevent two main issues that can disturb the proper function of the system: partial or complete sensor failure. While the complete failure is easier to detect (no reaction from a certain sensor), the partial defect is even more dangerous for two reasons. On the one hand, it can cause the sending of false data to the clusterheads, causing false positives or negatives. On the other, the development of effective detection of partial sensor failures is more complex than a simple test for a sensor's reactivity. The following paragraphs describe the testing routines of the sensors in detail.

Code 2 Pseudocode of Fletcher's checksum

```

//Computation of Fletcher's checksum
integer i, sum1, sum2
byte flash[image_length]
sum1 = 0
sum2 = 0

for i from 0 to (image_length - 1) do
    sum1 = (sum1 + flash[i]) modulo 0xFF
    sum2 = (sum2 + sum1) modulo 0xFF
end for

//Binary complement of sum1 and sum2 as reference
check1 = 0xFF - ((sum1 + sum2) modulo 0xFF)
flash[image_length] = check1

//Binary complement of sum1 and check1 as test checksum
check2 = 0xFF - ((sum1 + check1) modulo 0xFF)
flash[image_length + 1] = check2

```

- GPS module: The GPS module is checked each time it is activated for time synchronization purposes. After its activation, the GPS module receives a timing signal after a defined amount of time (one tick each second). If the timing signal is not received in this timeframe, a time-out event occurred. This leads to an error of the GPS module, that is reported by an error flag to the command center *c*. If the next timing signal is received correctly in time, the error flag is cleared again.
- AMR sensor: Before each use of the AMR sensor inside of the vehicle detection module, it has to fulfill a calibration self-test mechanism. This test works by actively compensating the earth's magnetic field by sending current through coils. The test routine uses the calibration to find errors in the sensor, which can be found by comparing the current to a reference value. If the difference exceeds a predefined value, an error is reported by setting an error flag. The error flag can be removed again if a following calibration returns an acceptable value.
- Acceleration sensor/accelerometer: The acceleration sensor located on the security module of the node is a key element in the tamper resistance concept of the sensor nodes. The test for this sensor is periodically triggered and uses an internal self-test algorithm of the security module. The test uses the apparent gravity to measure if the acceleration sensor is working correctly. If the self-test fails, which is the case if the difference of the current test result and the reference value stored in the test-algorithm exceeds a certain threshold, the error is reported by an error flag that is sent to the clusterhead. Again, if the test is completed successfully afterwards, the error flag is revoked.
- Geophones: The test routine of the geophones could not be built upon an internal test routine. Therefore, the testing mechanism can only verify if a geophone, which is connected to a sensor node, is able to receive and send messages. This is done by sending a wake-up message to the geophone. If the device is answering that message with an verification message in a certain amount of time, it thereby confirms to be working in unimpaired condition. Again, if either the verification

message is not received by the sensor node in time or not at all, an error flag is set. It is revoked the next time a verification message is received within the expected time.

- PIR sensors: Due to the nature of the PIR sensor, there is no self-testing procedure available, due to the lack of an reference value. The only malfunction of a PIR sensor which can be detected is the case when it is always on, because during normal operation, the sensor automatically deactivates directly after being triggered.

Reporting of and Reaction to Errors

To ensure a controllable system behavior, it is not only necessary to detect hardware malfunctions and tampering attempts, but also to report and react to these issues in an appropriate way. Because of that, a multi-level reporting and logging system is realized. The real-time reporting fulfills two important tasks: On the one hand, any detected error is reported to the command center c , where the user can see and if necessary, react to any integrity failures in the network. These are reported using an status byte consisting of 8 signaling flags (*cf.* Figure 7) which is sent from every sensor node to its clusterhead ch_i with each heartbeat message.

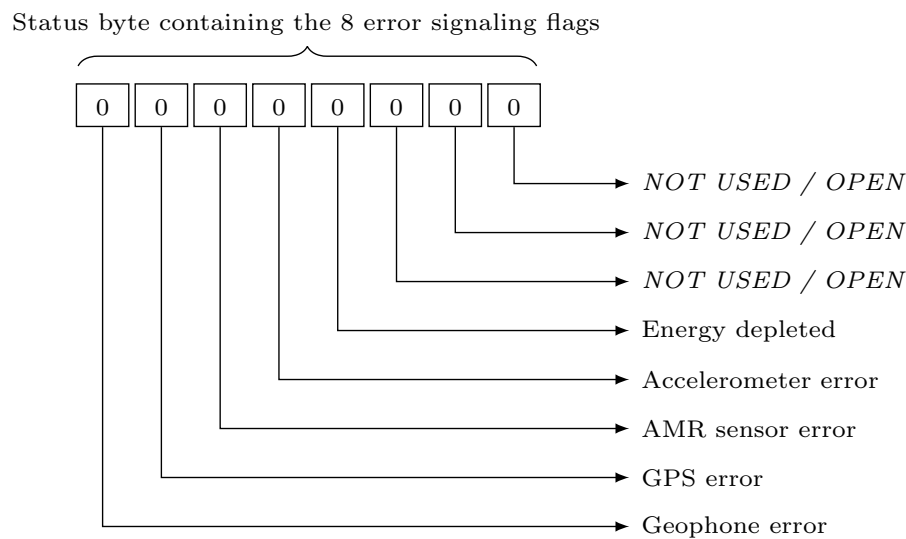


Figure 7. Status byte used for error flag signaling.

An error is indicated by setting the signaling flag from 0 to 1 at the appropriate position, as described in Section 3.3.2. If an error is sent from a sensor node n_{ij} , it is passed through the clusterhead ch_i to the command center c , where a message indicating the error is displayed to the user. On the other hand, certain errors have to cause an immediate reaction in the network, as an delayed reaction by the user would cause an overhead that is unacceptable. For example, a CPU or RAM error could endanger the correct function of the overall system by sending false data, leading to either false positive or false negative detection results. Another possibility would be that an impaired node sends too many messages leading to a reduced bandwidth availability for the other nodes. Therefore, nodes that do not pass the CPU or memory tests are prevented from joining the network without prior user interaction.

Moreover, the system includes a logging function, which is—in contrast to the real-time reporting—used to keep track of the events in the WSN over a longer period of time. This enables an user to assess actions in the surveyed area and detection results of the system in a time-delayed manner. The mechanism included uses a database to document all events coming to and all messages sent by the command center *c*.

A similar error indication signaling is used in the AFH mechanism. Here, a jammed channel is represented by setting the corresponding bit in the channel mask (*cf.* Figure 8) from 1 to 0. The jamming detection is done by the clusterheads, which distribute the information of the currently usable channels to their sensor nodes with each heartbeat message.

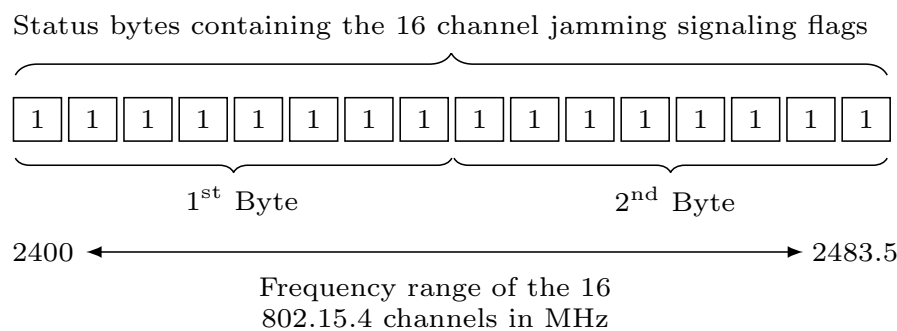


Figure 8. Channel mask used to signal jammed frequencies.

3.3.3. Energy Supply

As stated previously, the energy consumption needs to be highly restricted in sensor nodes, as the energy supply is very limited. This also applies to our system. In addition to the overall energy shortage, this issue is intensified by the usage scenario of the system: The security and reliability is one of the key features that must not be compromised. Because of that, a sudden failure of nodes due to energy drain has to be prevented. To do so, an early warning system is integrated into the sensor nodes that sends an alarm if the energy supply of a node drops lower than 10%, leaving the user enough time to change the node's power supply or replace it with another spare node. To measure the remaining energy, a energy consumption model is implemented that calculates the overall drained and remaining energy supply. This is done by measuring the time the system is in “awake” (sensors and node components activated) and “sleep” (sensors and node components deactivated) state. By doing that, it is possible to approximate the remaining system lifetime by subtracting the consumed energy (sum of energy used in both “awake” and “sleep” state) from the overall battery capacity. The remaining energy value is stored in the flash memory in addition to the RAM. This prevents miscalculations in the case of system restarts, as the remaining energy value is preserved in the flash memory. Tables 2 and 3 show the approximate energy demands of the different components and node types used in the WSN.

Table 2. Approximate energy demand of different components used.

Component	Awake Demand	Sleep Demand
Accelerometer	650 μ A	1 μ A
Single-PIR sensor	300 μ A	0 μ A
Multi-PIR sensor	900 μ A	0 μ A
GPS receiver	50 mA	0 mA
AMR sensor	40 mA	0 μ A
Core module	6 mA	40 μ A
FM transceiver	16 mA	0 mA
Geophone (GP)	10 mA	5 mA
IGEP module	600 mA	480 mA

Table 3. Approximate energy demand of different node types used.

Node Type	Awake Demand	Sleep Demand
Clusterhead	672 mA	480 mA
Node with	Multi-PIR, no GP	64 mA
	Multi-PIR, with GP	74 mA
	Single-PIR, no GP	63 mA
	Single-PIR, with GP	73 mA

3.4. Secure Detection, Localization and Classification Algorithms

In order to achieve a robustness and reliability suitable for long term surveillance, the detection, tracking and classification methods are implemented in a simple, yet effective, way in the hierarchical network. Figure 9 shows information flow of proposed detection, localization and classification algorithms.

- **Sensor node:** Located at the lowest level, sensor nodes are responsible for the acquisition of sensor data, with the equipped PIR sensors, magnetic sensors, Geophones and accelerometers. For PIR sensors this is a cone of a specific length whereas for all other sensors, this is a radial symmetric area around the sensor node with a certain radius. After the WSN is started, the detection algorithm first remains in a waiting state, until a significant amount of sensor events are registered, which indicates an object entering the area. If the amount of events remains low, it is assumed that these are caused by random noise or other sources, like wind. Additionally, it is required that there is a local accumulation of events. The amount of random noise depends on the sensor type as well as on other factors like the position of the sensor, the time of the day and the weather conditions. If the sensor data satisfies these conditions, it is inferred that an object is inside the surveillance area and an initial position is determined. Each sensor event suggests possible locations of the object.

- **Clusterheads:** The event data from neighboring sensor nodes is further processed at clusterheads. As each sensor node suggests a possible position for an object, at the clusterhead, the current position of the object is then determined by averaging the suggested positions of the sensor events while factoring in the previous position of the object. Afterwards, only sensor events in a vicinity of the current position of the detected object are considered, until the object leaves the surveillance area.
- **Command center:** As the control center of WSNs, the command center classifies different objects (person, person carrying metal object, car or unknown) by an analysis of the AMR and geophone events. Magnetic sensor events in the vicinity of the object suggest a car or a motorbike, and geophone events indicate footsteps, but there needs to be an accumulation of these events, which exceeds the noise level. Table 4 shows the classification matrix of events detected by different sensor types. For clarification, an example is given here: Assuming a Single-PIR event is triggered, at first, an entity is detected to be present in the WSN's vicinity, however a classification is not yet possible, therefore it is marked as "unknown". However, if the entity also triggers an AMR event, yet no geophone event, it is classified as a vehicle. Experiments show that the AMR sensors are surprisingly affected by wind (possibly due to movement of cables or the sensor node itself). An object in the surveillance area is considered to have left this area if the last known position is near the border of the area and further sensor events are below the detection level for some time.

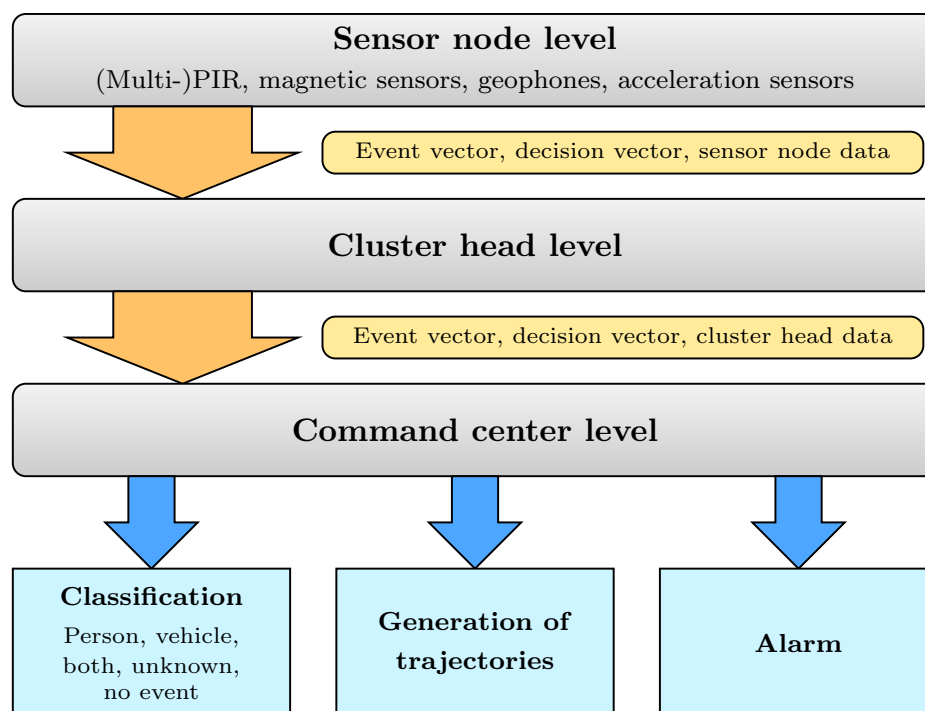


Figure 9. Information flow of detection, localization and classification algorithms.

Table 4. Classification matrix of events by different sensor types.

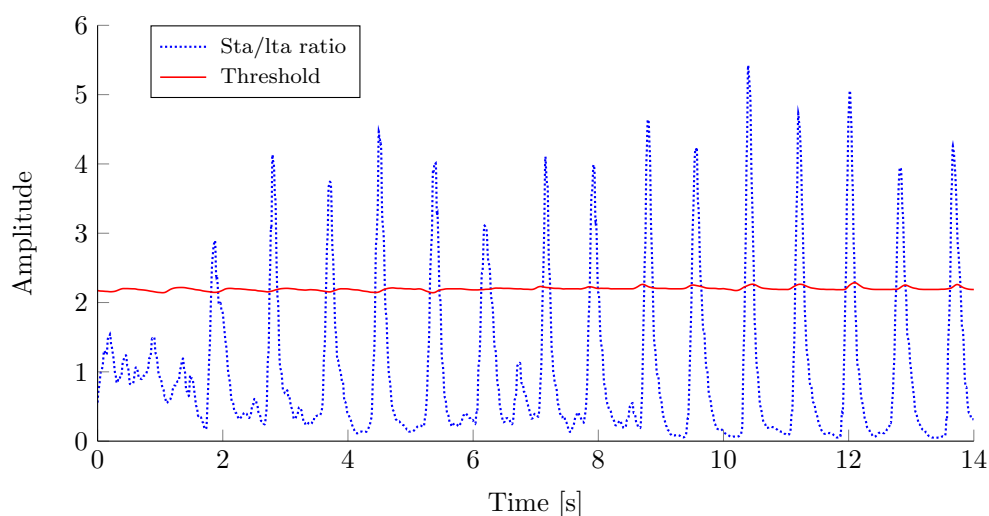
Component	Event Detected	
	Yes	No
Single-PIR	Person/vehicle	—
Multi-PIR	Person/vehicle	—
Long Range-PIR	Person/vehicle	—
AMR sensor	Vehicle	—
Geophone (GP)	Person	—

4. Sensor Evaluation

To achieve the desired quality of the surveillance, the utilized sensors need to be analyzed, since datasheets often do not provide information in the way or detail required. Additionally, the preprocessing algorithms of the geophones and PIR sensors need to be evaluated and characterized. In the following an excerpt of the sensor related experiments is given.

4.1. Geophones

In order to be able to detect intruders and therefore to provide a basis for tracking a robust algorithm is required which filters the incoming signal at the geophone to isolate those representing human footsteps. To overcome the problem of varying environmental conditions (mostly soil quality and humidity as well as ground coverage with plants) and of disturbing seismic noise, a procedure with an adaptive threshold is developed and implemented. It is based on the so-called “sta/lta-picking” which is used in earthquake location scenarios [20]. The decision whether there is an event is made by comparing the ratio of a short and a long time average of the seismic signal with an empiric constant. By introducing an adaptive constant an accommodation to varying conditions is achieved. An example is shown in Figure 10.

**Figure 10.** Results of the geophone event picking algorithm.

The output event of a geophone reliably signals a person within the detection range. By calculating the geometric center of gravity of one or more geophones reporting events with the same timestamp, it is possible to locate (and track) an intruder within the sensor network.

4.2. PIR Sensors

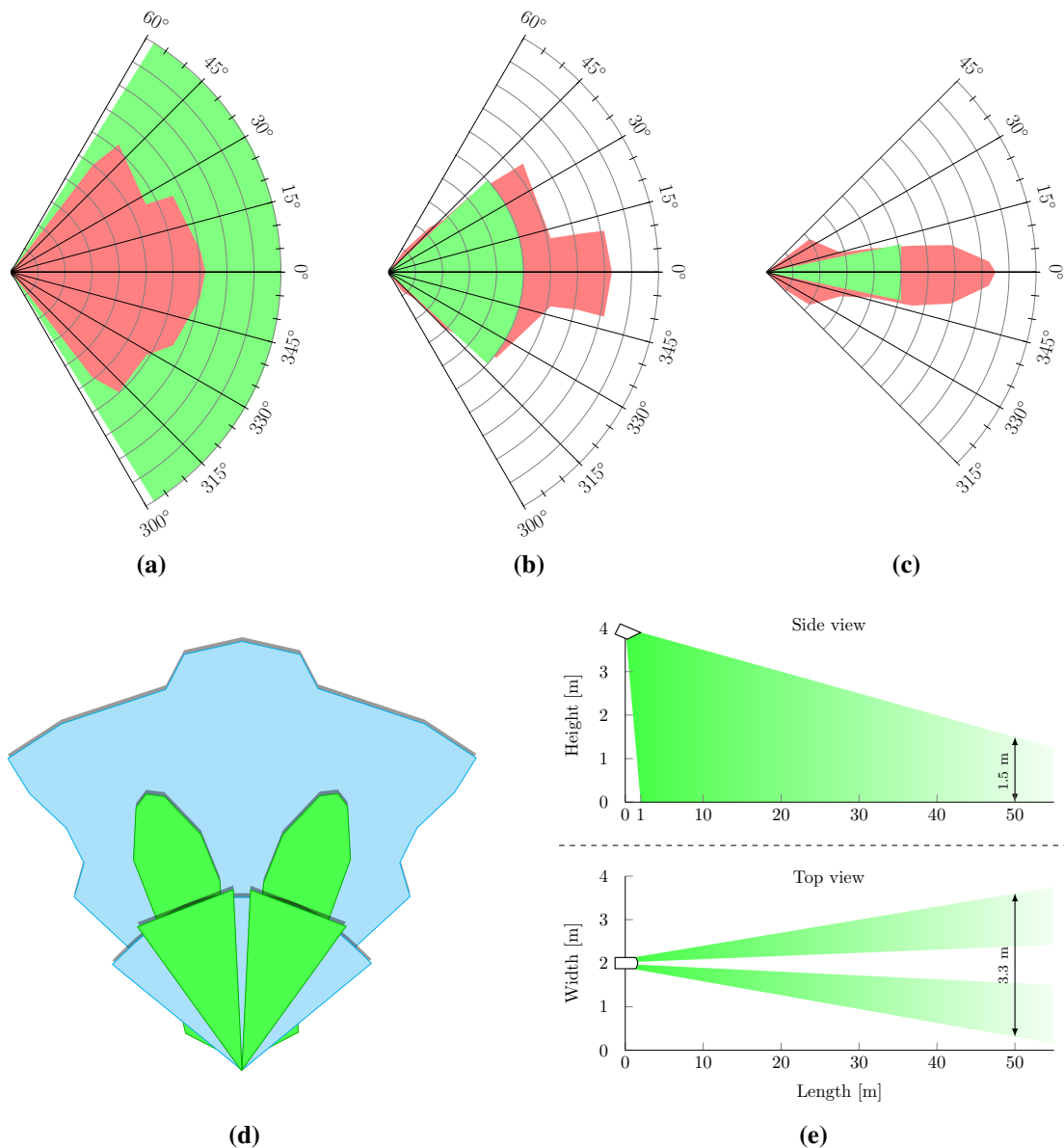


Figure 11. Experimental evaluation of the detection ranges of the different PIR sensor types. Green marks the detection area specified in the datasheet and red the actual detection area for walks with normal speed. (a) AMN34111 (Single). (b) AMN31111 (Multi). (c) AMN33111 (Multi). (d) Multi-PIR. (e) Long-Range-PIR: IS392.

For the PIR sensors, the region in which objects produce sensor events must be known in order to perform a localization of intruding objects. It is common to conduct simple walking and driving tests for this purpose, for which a path is defined and repeatedly driven, or walked, respectively. In Figure 11 the

results for normal walking are given. Green marks the detection area specified in the datasheet, while red marks the region in which sensor events actually occur in practice. In Figure ?? the joint detection region of a Multi-PIR sensor node is visualized. Here green marks the region where the two spot type PIR sensors (AMN33111) are sensitive and blue the region of the standard PIR type (AMN31111).

In Figure 12, the exemplary behavior of a Multi-PIR sensor node is shown for a walk-by test in which a person crosses the detection area from right to left with normal walking speed and vice versa. It has been found that the direction of the object can be determined reliably simply using the 3-PIR sensors included in a Multi-PIR sensor node. To do so, the time at which each PIR sensor activates is measured and compared. This is not possible using the Single-PIR sensors, but promises more accurate localization and tracking of objects.

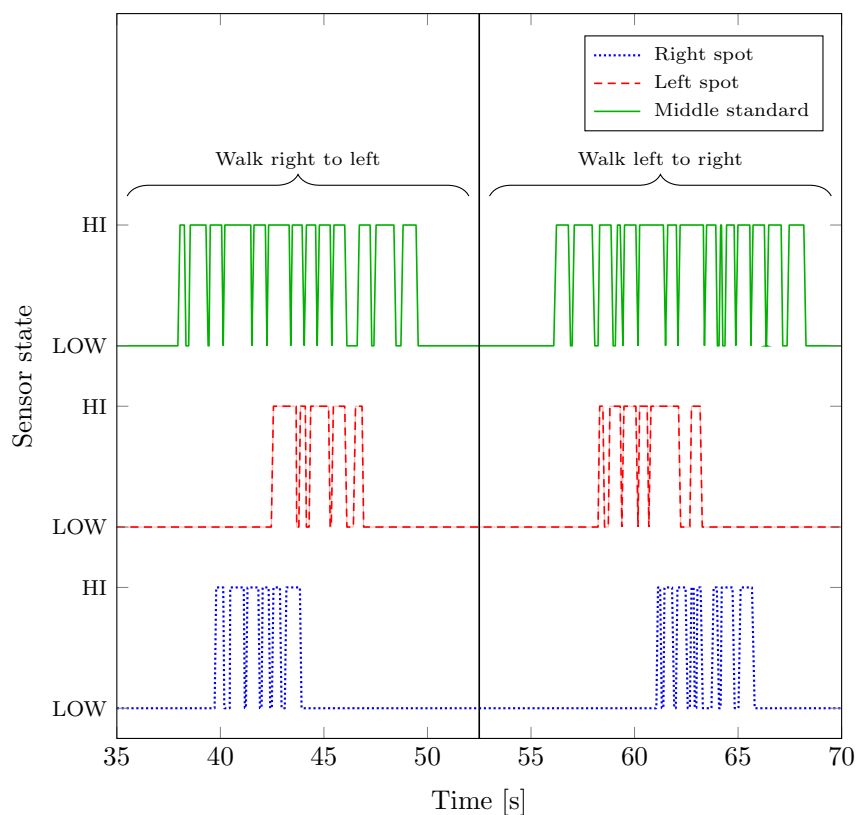


Figure 12. Typical PIR sensor activity of a Multi-PIR sensor node when an object passes the node from right to left and vice versa.

4.3. AMR Sensors

For the vehicle detection based upon the AMR sensors, the activation procedure, the sensor range and detection algorithms were tested prior to the implementation of the final system. The according test setup is shown in Figure 13.

Upon a sensor event from its PIR sensor, the sensor node sends a wireless message to the two AMR sensor nodes, simulating a delay as it would occur in the final installation, where the communication would work via the clusterhead, instead of directly between the sensor nodes. Upon reception of that

message, the AMR nodes activate their vehicle detection modules, de-gauss and calibrate the sensors to compensate for the static earth magnetic field, start sampling the two sensor module channels, and forward the live data to a forth sensor node that is connected to a PC to record timestamps and sensor data for all three nodes.

Two sensor nodes with a vehicle detection module are positioned at opposite sides of a road in a distance of 8 m from each other. A sensor node with a PIR sensor is placed next to the road 15 m before the AMR nodes. A vehicle then passes the installation at speeds of 50 km/h or 5 km/h, respectively.

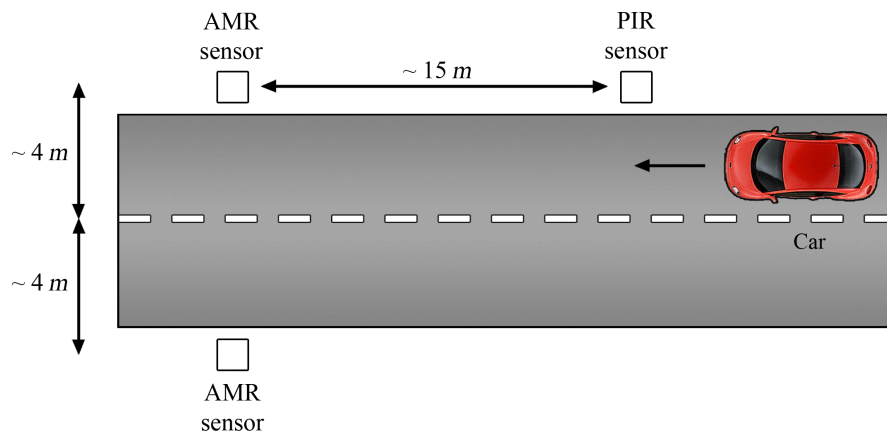


Figure 13. Field test of the AMR sensor.

Figure 14 shows the sensor readings of one of the AMR nodes for a vehicle speed of 50 km/h. The time axis of the diagram is set to start at 0 at the time when the PIR sensor event occurred. The y-axis shows the sensor signal as ADC digits, where 1 digit represents 0.59 mV or a field change of 786.2 mV/(kA/m). It is visible that the sensor signal starts around $t = 0.2$ s, as the calibration process commences before that.

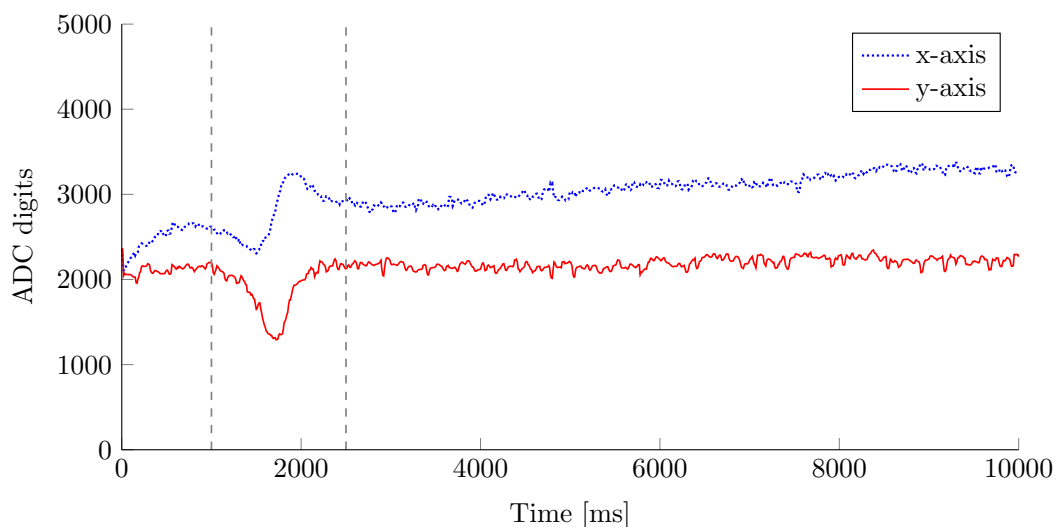


Figure 14. Sensor data with fast car movement (50 km/h).

The AMR sensor starts up and calibrates fast enough to be in normal operation by the time the vehicle passes by the sensor (characteristic signal shape between $t = 1$ s and $t = 2.3$ s). Consequently, the system would still work properly if the PIR sensor is placed at a distance of only 3 m to the AMR sensors.

Figure 15 shows the sensor readings of one of the AMR sensors for a vehicle speed of 5 km/h. As expected, the signal occurs much later, between $t = 9$ s and $t = 16$ s. Here, the challenge is rather to develop a detection algorithm that detects vehicles from both the signal patterns that result from 5 km/h and 50 km/h, while still being robust against the signal drift that occurs even though no vehicles passes.

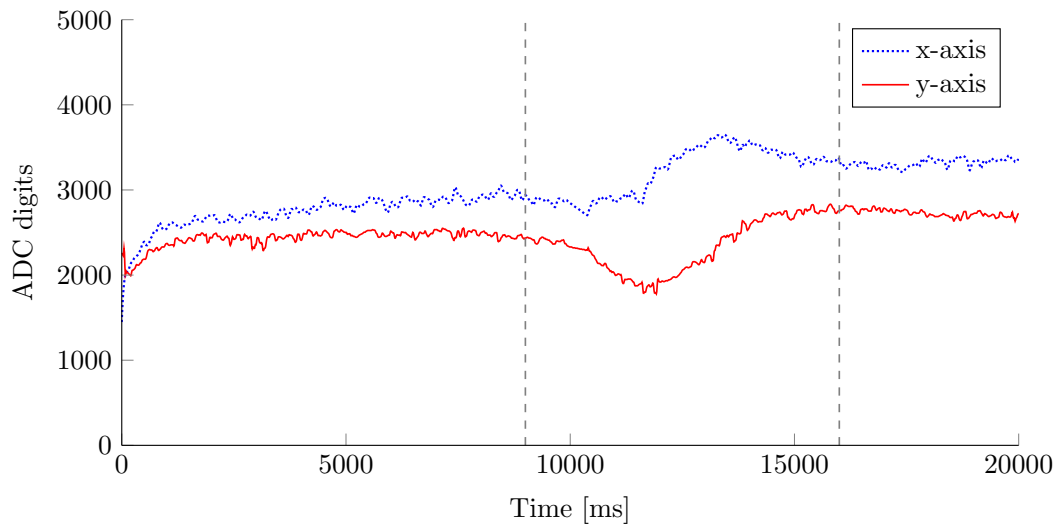


Figure 15. Sensor data with slow car movement (5 km/h).

5. Network Implementation and Assessment

In this section, the tests performed to evaluate the overall system performance in the areas of communication, object detection as well as security and safety are described. This is done by two different testing methods: tests in a laboratory environment on the one hand and live field trials under working conditions on the other. The testing methodology to verify the different parts of the sensor network is described in the following.

5.1. Laboratory Tests

5.1.1. Functional Safety

To ensure the proper function of all critical system components, the functional safety tests are conducted in a laboratory test to ensure a controlled environment. The assessment includes the manipulation of all protected sensors and a test for energy drainage, that is run to verify the expected life time of the sensor nodes. Tests for memory errors in RAM or flash, or CPU failures are not done, as causing such errors can only be achieved by either willingly damaging the hardware or by environmental factors (like radiation) that can only be simulated using highly specialized equipment that is not available in a normal laboratory. A detailed listing of the tests is given in Table 5.

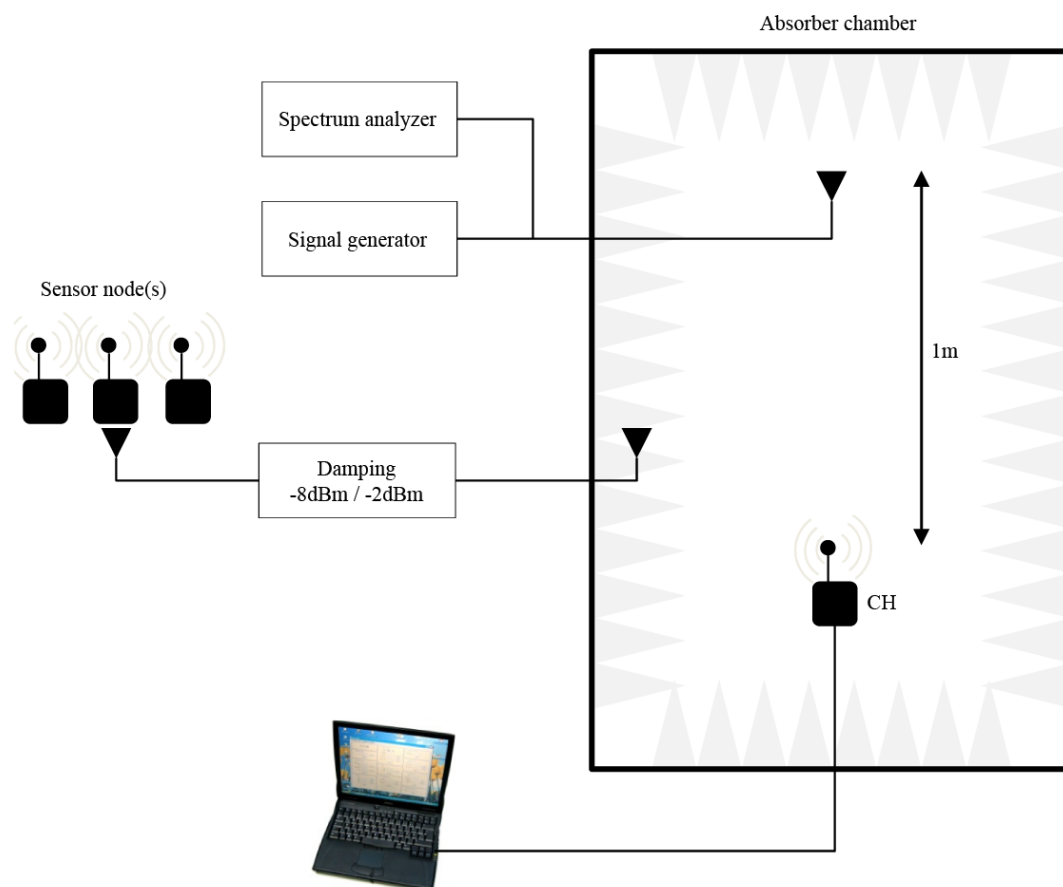
Table 5. Functional safety tests details.

#	Function	Action Performed	Time
1	AMR	Removal of module	Before start
2	AMR	Removal of module	During operation
3	Acceleration	Removal of module	Before start
4	Acceleration	Removal of module	During operation
5	Geophone	Disconnection of cable	Before start
6	Energy supply	Battery draining	During operation

On each sensor module, a connection plug is located in the upper right area of the module. In the tests, the module-removal is done by pulling the module off the rest of the sensor node stack. The geophone is disconnected by pulling the connection-cable and thereby interrupting the connection with the sensor node.

5.1.2. Adaptive Frequency Hopping and Communication Robustness

AFH mechanism is also verified in a separate laboratory test. The test arrangement for the different assessments is shown in Figure 16.

**Figure 16.** Adaptive frequency hopping and communication robustness test setup.

As depicted in Figure 17, the sensor node part of a clusterhead is placed inside of an absorber chamber and linked to a laptop via a serial connection to enable the logging of the test results. Additionally, a FM jamming signal emitter is placed at a fixed distance of 1 m inside the chamber. The jamming signal's frequency can be remotely configured from the outside. To measure the signals inside the absorber chamber, a spectrum analyzer is used.



Figure 17. Absorber hall with jamming device and clusterhead.

The signals of several sensor nodes—which are located outside of the absorber chamber—are routed through an antenna to the inside, simulating a distance of approximately 10 m between sensor nodes a clusterhead by damping the signal. In all tests, the sensor node(s) send a 10 Byte long packet every 50 ms (with an additional sending jitter of 10 ms to avoid synchronization between the nodes), leading to an overall data rate of 200 Byte/s. As the AFH switches between different channels every 250 ms, 5 packets are sent on each channel. A jammed channel is assumed if the data receive rate drops below 75% in the tests.

During the tests, different channels are jammed by an FM jammer. It operates at a bandwidth of 1 MHz around the center frequency of the according channel at an output power of -50 dBm with a NF-frequency of 400 Hz.

5.2. Field Trial

In the course of the system development, two field tests of the whole WSN were conducted. Figure 18 shows an illustration of the first testing terrain painted by the Spyglass application. The first assessment consisted of 100 sensor nodes and 10 clusterheads organized in 10 clusters, where one cluster was reserved for energy consumption monitoring. The surveillance area was a field with a path and had a maximum width of 125 m and a length of 50 m at the largest extend. The nodes were distributed arbitrarily in distances of 5 m to 10 m from each other. After this initial setup, the sensor clusters were defined by joining 10 sensor nodes n_{ij} , $1 \leq j \leq 10$ nearest to ch_i together in one cluster cl_i . After the

WSN was fully configured, several aspects of the whole system, and their interactions, had to be verified: First, the general system functions, like network connectivity, message sending, relaying and receiving as well as displaying of system events at the command center. Second, the detection, localization and classification had to be assessed and third a security evaluation to test the WSN's security and safety features was performed.

In the second field test of the WSN a total number of 17 sensor nodes partitioned in 2 clusters were utilized. The surveillance area was a small field surrounded by bushes and buildings with a path going through it. Again, a set of walking and driving detection test were performed.

A complete list of the test scenarios performed during the first and second field trial, including both detection and security tests, is given in Table 7.

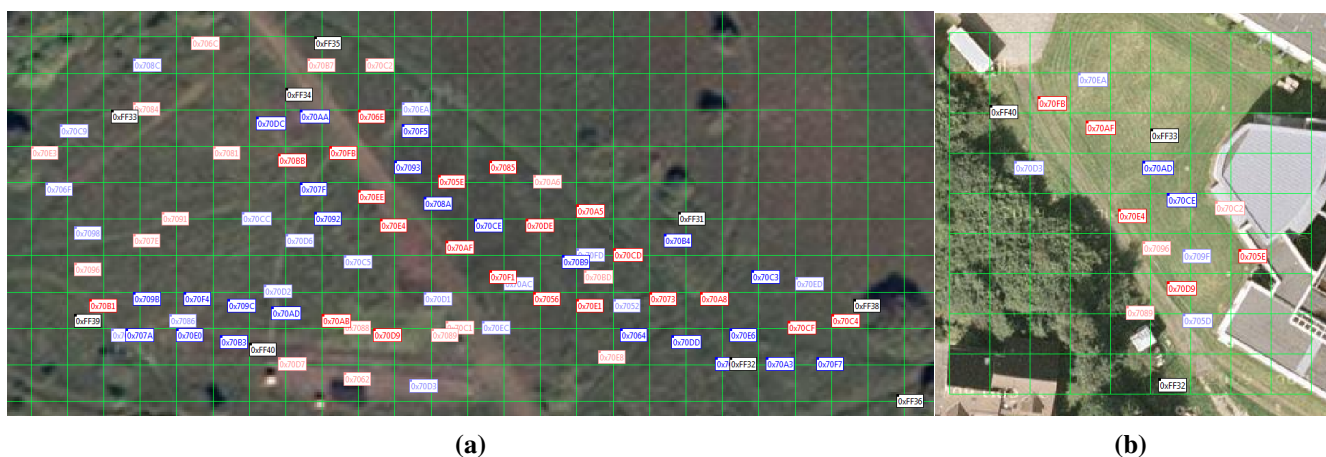


Figure 18. Spyglass renderings of the maps for the field tests performed within the scope of the system development (grid size is 5 m \times 5 m) (a) Large field test; (b) Small field test.

6. Results

6.1. Laboratory Tests

6.1.1. Adaptive Frequency Hopping and Communication Robustness

Figure 19 below shows the results of the AFH communication tests. It depicts the sequence of channel jamming, jamming detection and the closing of channels by the clusterhead, as well as the amount of data from the three sensor nodes employed during testing arriving at the clusterhead.

At $t = 34$ s, the jammer blocks channel 3. At $t = 37$ s, $t = 41$ s and $t = 45$ s, the clusterhead tries to send data using the blocked channel. As a result, the data sent on that channel is lost, the arriving data rate drops, and the clusterhead detects the jamming. Consequently, it closes channel 3 at $t = 45$ s (cf. “number of closed channels” goes to 1 in Figure 19). After that, the data rate stabilizes again.

The same procedure happens from $t = 68$ s to $t = 76$ s with channel 8 and from $t = 89$ s to $t = 97$ s with channel 12.

The fact that the data rate stabilizes shortly after the jamming of a channel starts shows that the AFH scheme implemented in the system effectively allows to detect channel jamming, and that the channel

closing algorithm significantly helps to preserve communication despite of jamming attacks.

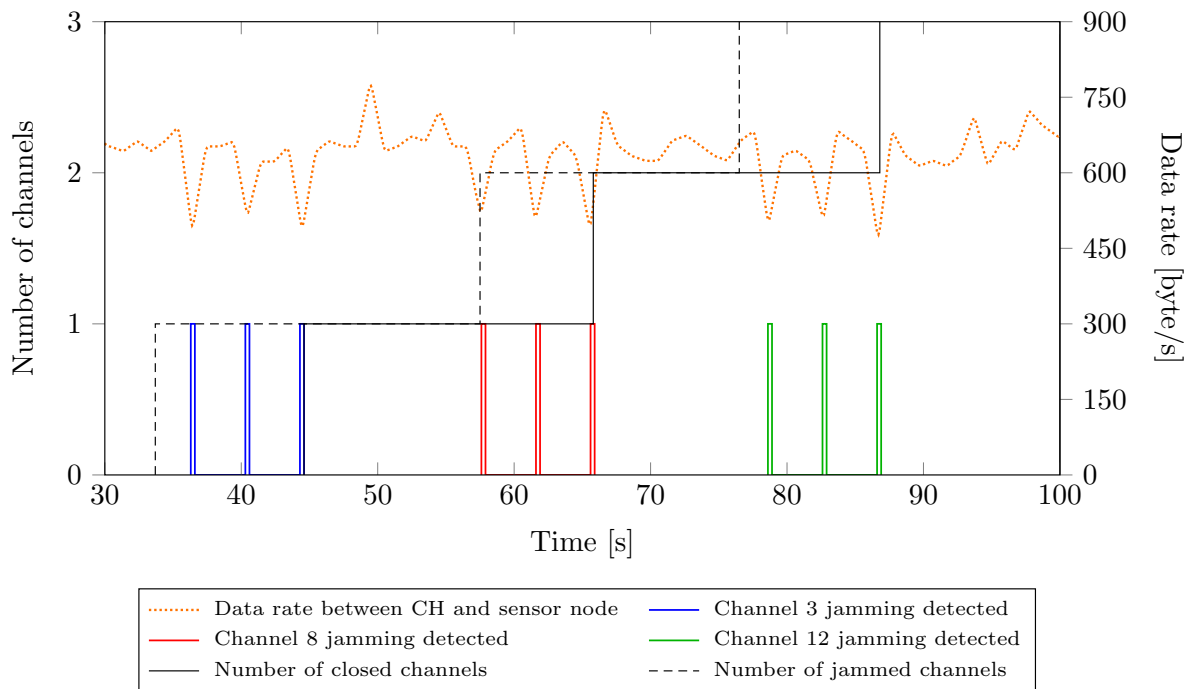


Figure 19. Events occurred during the AFH and jamming tests.

6.1.2. Functional Safety

The functional safety tests described in Table 5 all show the expected results. As stated in Section 5.1, the tests are done by logging the debug messages sent by the sensor nodes. This is visible in Code 3, where the accelerometer is removed during the operation of an arbitrarily chosen sensor node. As line 4 of the log shows, a hardware failure is detected.

Code 3 Log of the test #4

```
[2011-09-13 16:25:55.669] pir2 down at 27:14.754
[2011-09-13 16:25:55.669] pir2 up at 27:14.757
[2011-09-13 16:25:55.825] pir0 up at 27:14.899
[2011-09-13 16:25:55.825] HW-FAILURE DETECTED. ID: 3
[2011-09-13 16:25:55.872] pir0 down at 27:14.952
[2011-09-13 16:25:55.981] HW-OK DETECTED. ID: 2
```

The *ID* of the error is the position of the error flag in the status byte described in Figure 7. The results of the other sensors behave in a similar way to the assessment of the accelerometer. As an exhaustive listing of all results would exceed the length of this paper, as a second example, the evaluation of the AMR sensor can be seen in Code 4.

Code 4 Log of the test #2














```

[2011-09-21 17:20:40.387] pir0 down at 27:01.567
[2011-09-21 17:20:40.402] calibrate AMR
[2011-09-21 17:20:40.418] JI2C: No ack rx
[2011-09-21 17:20:40.433] AMR flip
[2011-09-21 17:20:40.449] JI2C: No ack rx
[2011-09-21 17:20:40.449] AMR flip
[2011-09-21 17:20:40.465] JI2C: No ack rx
[2011-09-21 17:20:40.480] AMR flip
[2011-09-21 17:20:40.480] JI2C: No ack rx
[2011-09-21 17:20:40.496] AMR flip
[2011-09-21 17:20:40.511] HW-FAILURE DETECTED. ID: 2

```

Another part of the evaluation is the test of the tamper resistance feature based on the accelerometer. To trigger this manipulation alarm, the sensor node is lifted off the ground and then shook several times. The result is a visible indicator (“Accelerometer event” in Table 6) at the command center, signaling a possible manipulation.

Table 6. Event symbols and classification colors in Spyglass.

Symbol	Meaning	Symbol	Meaning
	Geophon event		AMR event
	Longrange-PIR event		Multi-PIR event
	Accelerometer event		Single-PIR event
	Node communication issue		Node failure
	Energy drain alert		
Color	Classification	Color	Classification
	Human		Vehicle
	Human with metal object		Unknown

6.2. Field Trial

To verify the functions assessed in the field trials, multiple verification techniques were used. For the detection, localization and classification, a Mobotix IP-camera was used in combination with the Spyglass user interface. By doing so, it is possible to check if objects were on the one hand detected and localized, on the other if the classification works correctly and the real-time requirement of the system is fulfilled.

As an example, Figure 20 shows one of the walking tests done during the second field trial. As the figure depicts, the person walking on the grass was detected and classified correctly, which is indicated

by the green field in the middle of the grid. Similar results were also achieved for other walking styles, like crawling or running, as well as tests with other trespassing objects, like e.g., cars (the classification changed in those tests respectively).



Figure 20. Example of a walking test during the second field trial.

The field trials were also used to optimize the system parameters, especially the calibration of the sensor sensitivity posed a challenge during the first assessments due to the complex processing and classification applied to the sensor events. The geophones are particularly sensitive to parameter changes, as they support a very broad range of sensitivity values. These values have to be adjusted in a way that they are on the one hand reactive enough to detect even a crawling human trespasser, while on the other they are not overreactive, which would lead to a too high number of geophones responding to seismic events at once. This, in turn, would disturb the localization capabilities of the overall system as the geophone reaction would not be localized enough.

In the energy endurance test (Test #33 in Table 7) performed during the first field trial, it became clear that a glitch in the clusterhead software hindered the GPS module to enter standby mode. This caused a significant energy drain, resulting in a severely shortened lifetime of only six days. This problem was fixed by implementing an adaptive standby algorithm, which compares the internal clock $T_{internal}$ of the clusterhead with the GPS time signal T_{gps} in increasingly large time intervals T_{check} . As long as the time difference $\Delta T = |T_{internal} - T_{gps}| < 1$ ms, T_{check} is doubled every time the time difference test is run, starting at 20 s and ending at a maximum of 3600 s. As a result, the lifetime of the clusterheads was extended to 17 days on average. The energy consumption of the sensor nodes was also assessed. In the field trials, which lasted for more than two weeks, no energy-related unavailability could be discovered. After an extensive analysis with energy consumption models (Section 3.3.2), an average node lifetime is expected to be 28 days (depending on the sensors used and the amount of registered events).

Additionally, the detection tests showed that it is possible to detect, locate and classify persons and vehicles (cars & motorbikes) with high confidence (Tests #1–3 and #12–29 in Table 7). Furthermore, it was evaluated if a classification of a person carrying metal objects is possible, which would be especially useful to detect weapons like firearms. While the detection works with large ferromagnetic objects, like e.g., fire extinguishers or crowbars, as long as the distance between the object and the sensor node is ≤ 2 m, it is not possible for firearms. This is due to the low mass of ferromagnetic parts in modern

firearms, resulting in a very low amplitude of the AMR sensor's readings, which does not trigger an event. All classification grid colors and symbols used in Spyglass are shown in Table 6.

Table 7. List of performed field trial tests.

Detection Test Scenarios					
Test #	Distance to Sensor Nodes	Terrain	Type of Trespasser	Employed # of Clusters	Penetration
1	2 m	Short grass	Person	1	Yes, shortest path
2	2 m	High grass	Person	1	Yes, shortest path
3	2 m	Stony ground	Person	1	Yes, shortest path
4	4 m	Short grass	Person	1	Yes, shortest path
5	4 m	High grass	Person	1	Yes, shortest path
6	4 m	Stony ground	Person	1	Yes, shortest path
7	10 m	Short grass	Person	1	Yes, shortest path
8	10 m	High grass	Person	1	Yes, shortest path
9	10 m	Stony ground	Person	1	Yes, shortest path
10	Variable	Short grass	Person carrying metal object	Arbitrary	No
11	Variable	Short grass	Person carrying metal object	Arbitrary	Yes
12	Variable	Stony ground	Car	Arbitrary	No
13	Variable	Stony ground	Car	Arbitrary	Yes, 10 km/h
14	Variable	Stony ground	Car	Arbitrary	Yes, 40 km/h
15	Variable	Stony ground	Person	Arbitrary	Yes, running
16	Variable	Short grass	Person	Arbitrary	Yes, duck gait
17	Variable	Short grass	Person	Arbitrary	Yes, creeping
18	Variable	Short grass	Person	1	Yes, zigzag pattern
19	Variable	Short grass	Person	2	Yes, zigzag pattern
20	Variable	Short grass	Person	1	No
21	Variable	Short grass	Person	2	No
22	Variable	Stony ground	Bicycle	Arbitrary	Yes
23	Variable	Stony ground	Motorcycle	Arbitrary	Yes, slow
24	Variable	Stony ground	Motorcycle	Arbitrary	Yes, fast
25	Variable	Stony road	Person	1	Yes, shortest path
26	Variable	Stony road	Person	2	Yes, running
27	Variable	Stony road	Car	Arbitrary	Yes, slow
28	Variable	Short grass	Person	Arbitrary	Yes, shortest path
29	Variable	Short grass	Person	Arbitrary	No, U pattern
Security Test Scenarios					
Test #	Terrain	Action Performed	Type of Trespasser		
30	Short grass	Shaking of a sensor node	Person		
31	Short grass	Removal of a sensor node out of the test field	Person		
32	Short grass	Reactivation of a sensor node in the test field	Person		
33	Short grass	Measurement of the energy consumption of a cluster	—		

To assess the limitations of the system, its behavior in borderline cases was analyzed by running several tests that evaluated the detection and localization limits of the sensors. As the localization is mainly done by the PIR sensors and geophones, this scenarios simulated the failure of several nodes in the same area, leaving gaps of 4 m (Tests #4–6 in Table 7), or 10 m (Tests #7–9 in Table 7) respectively, between the sensor nodes. While a detection and localization of a trespasser was still possible at a sensor node distance of 4 m without a major impairment, this changed when the distance was increased to 10 m. Due to the limited range of the PIR sensors (~ 6.5 m, *cf.* Figure 11), in the test cases #7–9, where a 10 m distance was used, both the localization and classification capabilities suffered a severe impact. The network was however still able to identify that an object was present in the surveillance area.

To test the system's ability to detect unavailable sensor nodes (e.g., due to physical removal or damaged wireless interfaces), one test scenario included the removal and shutdown of a single node out of the sensor network (Test #31 in Table 7). The removal first resulted in a tampering alarm, because of the necessary physical manipulation of the node by removing it. As a wireless communication is prone to data loss, the indication of node unavailability was realized using two stages, which indicate a temporally limited or permanent loss of node availability. At first, the status of the node is set to "Node communication issue" (cf. Table 6), which is triggered if a node's heartbeat message is not received three times in a row. If a node resumes communication within 7 s after its status was set to "Node communication issue", it is reset to unimpaired status. If no heartbeat is received within 7 s, the node is marked as "Node failure" (cf. Table 6), meaning it is permanently unavailable. In the test, the described behavior could be observed.

Moreover, the varying weather conditions during the field trials, ranging from sunshine to thunderstorms, proved the system's adaptability and its resistance against hazardous environments.

7. Related Work

The presented work is manifold combining multiple research fields, e.g., communication security, functional safety but also energy efficient communication and the fusion of WSN data. Therefore, this section first covers the state-of-the-art in conventional (non-networked) solutions for secure infrastructures. Second, related work on systems that rely on wireless communication and work in critical environments is described. Finally, recent projects that partly cover the goals of our work are introduced and compared.

Conventional solutions to secure critical infrastructures range from very simple setups involving mechanical barriers, like doors, fences and trenches or security personnel [21] to complex—however non-networked—equipment, including e.g., passive infrared (PIR), sound and seismic sensors, or video surveillance. While these devices allow detecting certain influences, they lack the possibility to cooperatively analyze a situation [22], as no data transfer between them is possible.

However, over the last years, advances in the area of networked sensor technologies have opened up new application areas for these sensor devices [23]. With new technological paradigms, like the Internet of Things (IoT) or Automated Living, the use of WSNs is rapidly growing, as networked sensors are vital in these scenarios. However, as future technologies more and more depend on sensor networks, the requirements for these—now critical infrastructures—rise in a similar way, especially in the fields of security and safety. An already realized example in this area is the SmartSantander project [24], where more than 10,000 IoT devices are deployed to build a so-called "Smart City". In such scenarios, strict security requirements apply, as personal and critical information is transferred. While there are research efforts in the areas of access protection, secure reprogramming over wireless connections and secure communication, a complete security solution for a wireless network does not yet exist. This example already characterizes several important aspects of WSN security: Due to the usage in unsupervised or even hostile environments while performing critical tasks, it is important to provide measures ensuring integrity and confidentiality of data (communication security) and protection of the devices against malfunctions that either occur randomly or due to tampering [25,26].

The POmSe (“Personen- und Objektdetektion mit mobilen Sensoren”) project [27], evaluates the general applicability of WSNs for the protection of critical infrastructures and environments with a focus on the applicability for trespasser detection due to their inherent advantages compared to conventional security techniques (like e.g., security doors, fences, *etc.*). In contrast to our work, POmSe performs a general analysis of WSNs including the required security features in critical applications. However, the development, application and evaluation of a WSN is not included in the project, neither is a classification of trespassing objects performed.

A complementary project, named Personen- und Objekterkennung basierend auf Trittschall (POT) [28], evaluates the suitability of seismic sensors for the detection, classification and localization of humans and animals. It turns out that a reliable detection algorithm can be provided whereas the exact spatial localization of a person strongly depends on the environmental conditions as, e.g., the type and humidity of the surrounding soil.

There are also some other works, such as FleGSens [29], Line in the Sand [30], Border Sense [31] etc, leverage WSNs for border intrusion detection and surveillance [32]. FleGSens is able to detect movement in critical areas and considers authenticity and data integrity of alarm signals. Other security issues, like jamming and functional safety, are not addressed. FleGSens uses a flat network hierarchy, which limits system scalability due to message collision. The used detection algorithm only considers the position of trespassers and—in contrast to our work—does not include a classification algorithm. Other works can either detect, classify and (or) track metal and nonmetal objects or detect passing ships and their speeds. However, security issues, like security for network topologies and security for data communication, are not tackled. As argued in [32], few WSN-based mission-critical applications are really experimented and deployed. Compared with the previous works, the main difference is the implementation of a comprehensive security solution covering all necessary aspects, while still maintaining a high usability despite the high system complexity. To ensure its applicability for critical infrastructure monitoring, in this paper, the involved sensors are comprehensively evaluated and tested in both laboratory environment and live field trials. This paper improves our previous work [33] by implementing functional safety / self-protection functions and adding energy supply awareness. Further, this paper also includes laboratory tests for adaptive frequency hopping, communication robustness and functional safety.

8. Conclusions and Future Work

As this paper demonstrated, the developed system is a WSN solution that is capable of monitoring and securing critical infrastructures. To do so, it offers security by using different sensors to detect, locate, track and classify various types of trespassers while operating in a secure and safe way. This is done by employing several functional safety and security features which guarantee high levels of confidentiality, integrity and availability. In addition to the features described before, the system operates in real-time using its efficient detection algorithms and network hierarchy. It enables its user to keep track of the events inside the sensor network by logging all events in a database for later analysis and, at the same time, reporting it in real-time to the command center. Moreover, the system uses a scalable, hierarchic network structure making it easily extendable for a wider area, if necessary. To adapt the system to

changing usage conditions, it is possible to dynamically reconfigure the whole system in the field by using an over-the-air-programming (OTAP) function. This unique combination of features makes the system ideal for unattended operations in hazardous environments.

While the described features already enable a secure detection of trespassing objects, future work could go in three directions. One possibility would be to include additional hardware in the concept, e.g., IP-cameras, to document trespassing. These devices would be integrated as a type of additional sensor into the existing sensor nodes and be activated if an object is detected by the other sensors. Of course, these new nodes would need to be carefully analyzed, especially their energy consumption and network traffic generation would need to be carefully balanced with the achievable visual quality. Another possible method would be to treat the sensor nodes with visual capabilities separately (and not integrate them into the existing network hierarchy). The visual data would e.g., be sent directly to the command center using WLAN. This would on the one hand save energy and network bandwidth, by bypassing the clusterheads and IEEE 802.15.4 network altogether, however, the hierarchical network structure would be broken using that concept. Therefore, a critical analysis would need to be performed before deciding which strategy to choose.

Another option would be to extend the software used for object detection. Currently, the software is able to monitor and detect the trespassing of a single target in the surveillance area (“Single target”). In the future, this software could be extended to enable the monitoring of multiple targets at once (“Multi target”). In addition, the current trespassing decision method offers a binary approach: either the target is trespassing or not. This method could be enhanced by adding a risk estimate that is based on the trespasser’s speed and direction. Another promising direction is to extend the current system by integrating mobile sensors. There are various forms of these devices, such as sensor-equipped cell phones, moveable sensor nodes or other mobile, networked devices using appropriate sensors. This would have the benefit that mobile sensors can provide context awareness based on the location and trajectory of a trespassing entity.

Acknowledgments

The research leading to the results presented in this paper was supported by the “Security, Education and Competence for Bavarian IT” (SECBIT) project, which is co-funded by the European Union’s European Regional Development Funds – Regional Competitiveness and Employment, the European Commission’s Project No. 608090, HyRiM (Hybrid Risk Management for Utility Networks) under the 7th Framework Programme (FP7-SEC-2013-1), as well as the project MOVEDETECT supported by the German Federal Office for Information Security (BSI—Bundesamt für Sicherheit in der Informationstechnik).

Author Contributions

Michael Niedermeier and Hermann de Meer coordinated the project leading to the results in a technical and organizational manner. They also contributed the security and safety solution utilized by the WSN. Michael Koch performed all technical and practical work related to geophones; Stefan Fischer and Dennis Pfisterer were responsible for the Wisebed and Spyglass applications; Klaus Hartmann

and Benjamin Langmann investigated and chose the different sensor types employed in the sensor nodes. They also provided major contributions in the sensor data aggregation and interpretation. Carsten Buschmann provided insight knowledge on sensor nodes and their communication abilities. The writing of the paper was mainly performed by Michael Niedermeier and Xiaobing He, with editorial support by Hermann de Meer.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Estrin, D.; Govindan, R.; Heidemann, J.; Kumar, S. Next Century Challenges: Scalable Coordination in Sensor Networks. In Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99), Seattle, WA, USA, 15–19 August 1999.
2. Asada, G.; Dong, M.; Lin, T.S.; Newberg, F.; Pottie, G.; Kaiser, W.J.; Marcy, H.O. Wireless integrated network sensors: Low power systems on a chip. In Proceedings of the 24th IEEE European Solid State Circuits Conference (ESSCIRC 98), The Hague, The Netherlands, 22–24 September 1998.
3. SpyGlass, a Modular and Extensible Visualization Framework for Wirelesssensor Networks. Institute of Telematics, University of Lübeck, Lübeck, Germany, 2006. Available online: <https://github.com/itm/spyglass> (accessed on 23 October 2015).
4. coalesenses GmbH. iSense Core Module 3 Data Sheet, 2010. Available online: http://www.coalesenses.com/download/data_sheets/DS_CM30X_1v3.pdf (accessed on 23 October 2015).
5. coalesenses GmbH. Gateway Module Data Sheet, 2010. Available online: http://www.coalesenses.com/download/gateway_module_data_sheet_1v2.pdf (accessed on 23 October 2015).
6. coalesenses GmbH. GPS Module Product Brief, 2010. Available online: http://www.coalesenses.com/download/product_briefs/ProductBriefGpsModule.pdf (accessed on 23 October 2015).
7. Buschmann, C.; Pfisterer, D. iSense: A Modular Hardware and Software Platform for Wireless Sensor Networks. In *Technical Report, 6. Fachgespräch Drahtlose Sensornetze der GI/ITG-Fachgruppe Kommunikation und Verteilte Systeme*; RWTH Aachen: Aachen, Germany, 2007.
8. NXP Laboratories UK. Data Sheet: JN5148-001 IEEE802.15.4 Wireless Microcontroller, 2010. Available online: http://www.jennic.com/files/product_briefs/JN-DS-JN5148-1v6.pdf (accessed on 23 October 2015).

9. IEEE-SA Standards Board. *IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan area Networks Specific Requirements-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*; Technical Report; IEEE Computer Society: New York, NY, USA, 8 September 2006.
10. The Federal Information Processing and Announcing. *Announcing the Advanced Encryption Standard (AES)*; FIPS 197; Federal Information Processing and Announcing: Gaithersburg, MD, USA, 26 November 2001.
11. SIEMENS. IS392, IS392H Outdoor Passive Infrared Detector, 2005. Available online: http://www.ntcbpl.ru/pdfinfo/Data-sheet_IS392_en.pdf (accessed on 23 October 2015).
12. Panasonic. MP Motion Sensor (AMN1,2,4), 2004. Available online: <http://datasheet.octopart.com/AMN34111-Panasonic-datasheet-110118.pdf> (accessed on 23 October 2015).
13. Input/Output, Inc. SM-24 Geophone Element, 2006. Available online: <http://cdn.sparkfun.com/datasheets/Sensors/Accelerometers/SM-24%20Brochure.pdf> (accessed on 23 October 2015).
14. coalesenses GmbH. Vehicle Detection Module Data Sheet, 2010. Available online: http://www.coalesenses.com/download/product_briefs/ProductBriefVehicleDetectionModule.pdf (accessed on 23 October 2015).
15. Coulson, G.; Porter, B.; Chatzigiannakis, I.; Koninis, C.; Fischer, S.; Pfisterer, D.; Bimschas, D.; Braun, T.; Hurni, P.; Anwender, M.; *et al.* Flexible experimentation in wireless sensor networks. *Commun. ACM* **2012**, *55*, 82–90.
16. Mpitziopoulos, A.; Gavalas, D.; Konstantopoulos, C.; Pantziou, G. A survey on jamming attacks and countermeasures in WSNs. *IEEE Commun. Surv. Tutor.* **2009**, *11*, 42–56.
17. Whiting, D.; Housley, R.; Ferguson, N. Counter with CBC-MAC (CCM), 2003. Available online: <http://tools.ietf.org/html/rfc3610> (accessed on 23 October 2015).
18. Lashkari, A.H.; Danesh, M.M.S.; Samadi, B. A survey on wireless security protocols (WEP, WPA and WPA2/802.11i). In Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009), Beijing, China, 8–11 August 2009; pp. 48–52.
19. Fletcher, J.G. An arithmetic checksum for serial transmissions. *IEEE Trans. Commun.* **1982**, *30*, 247–252.
20. Havskov, J. *Instrumentation in Earthquake Seismology (Modern Approaches in Geophysics)*; Springer Science + Business Media B.V.: Dordrecht, The Netherlands, 2004.
21. *Practices for Securing Critical Information Assets*; Critical Infrastructure Assurance Office: Gaithersburg, MD, USA, 2000.
22. Intanagonwiwat, C.; Govindan, R.; Estrin, D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 00), Boston, MA, USA, 6–11 August 2000; pp. 56–67.
23. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Comput. Netw.* **2002**, *38*, 393–422.

24. The SmartSantander Consortium. SmartSantander, 2010. Available online; <http://www.smartsantander.eu/> (accessed on 23.10.2015).
25. Roman, R.; Zhou, J.; Lopez, J. On the security of wireless sensor networks. In Proceedings of the International Conference on Computational Science and Its Applications—ICCSA 2005, Suntec City, Singapore, 9–12 May 2005; pp. 681–690.
26. Perrig, A.; Stankovic, J.; Wagner, D. Security in Wireless Sensor Networks. *Commun. ACM* **2004**, *47*, 53–57.
27. Bonitz, F.; Ghobadi, S.E.; Hartmann, K.; Hauff, H.; Herrmann, R.; Kargel, C.; Löprrich, O.E.; Heckmann, D.; Maisch, M.M.; Seidl, A.; *et al.* *Personen- und Objektdetektion mit mobilen Sensoren—Abschlussbericht zum Arbeitspaket 3 (Teil B)—Bericht zum Meilenstein 5*; Internal End Report; University of Passau: Passau, Germany, 2010.
28. Koch, M.; Hubert, C. *Personen- und Objekterkennung Basierend auf Trittschall (POT)*; Internal Technical Report; University of Passau: Passau, Germany, 2011.
29. Rothenpieler, P.; Krüger, D.; Pfisterer, D.; Fischer, S.; Dudek, D.; Haas, C.; Zitterbart, M. Flegsens—Secure area monitoring using wireless sensor networks. In Proceedings of the 4th Safety and Security Systems in Europe, Micro Materials Center Berlin at Fraunhofer Institute IZM and Fraunhofer ENAS, Potsdam, Germany, 4–5 June 2009.
30. Arora, A.; Dutta, P.; Bapat, S.; Kulathumani, V.; Zhang, H.; Naik, V.; Mirral, V.; Cao, H.; Demirbas, M.; Gouda, M. A line in the sand: A wireless sensor network for target detection, claffisication, and tracking. *Comput. Netw.* **2004**, *46*, 605–634.
31. Sun, Z.; Wang, P.; Vuran, M.C.; Al-Rodhaan, M.A.; Al-Dhelaan, A.M.; Akyildiz, I.F. BorderSense: Border patrol through advanced wireless sensor networks. *Ad Hoc Netw.* **2011**, *9*, 468–477.
32. Felemban, E. Advanced Border Intrusion Detection and Surveillance Using Wireless Sensor Network Technology. *Int. J. Commun. Netw. Syst. Sci.* **2013**, *6*, 251–259.
33. Langmann, B.; Niedermeier, M.; de Meer, H.; Buschmann, C.; Koch, M.; Pfisterer, D.; Fischer, S.; Hartmann, K. MOVEDETECT—Secure detection, localization and classification in wireless sensor networks. In *Internet of Things, Smart Spaces, and Next Generation Networking*; Balandin, S., Andreev, S., Koucheryavy, Y., Eds.; Springer Berlin Heidelberg: St. Petersburg, Russia, 2013; Volume 8121, pp. 284–297.