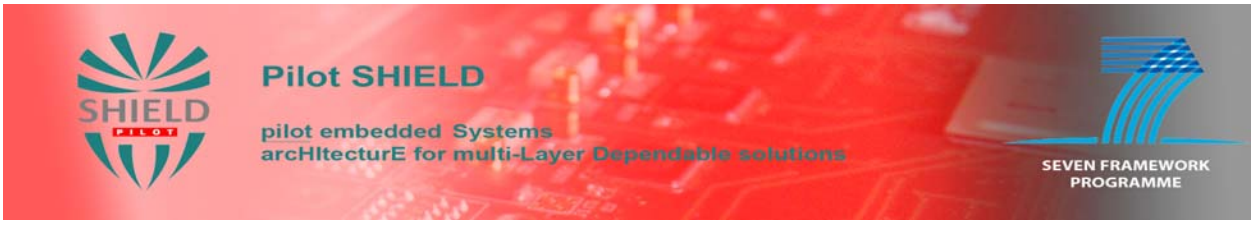

Project no: 100204

pSHIELD

pilot embedded **S**ystems arc**HI**tectur**E** for multi-**L**ayer **D**ependable solutions

Instrument type: Capability Project

Priority name: Embedded Systems / Rail Transportation Scenarios

SPD Network Technologies Prototype Report

Deliverable D4.2

Partners that contributed to the work:

MGEP, Mondragon Goi Eskola Politeknikoa, Spain
UNIGE, Università di Genova, Italy
CS, Critical Software, Portugal
ATHENA, Greece
THYIA, Slovenia

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2012)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Authors and Approvals

Authors		Date	Signature
Name	**Company**		
Roberto Uribeetxeberria	MGEP		
Iñaki Arenaza	MGEP		
Urk Zurutuza	MGEP		
Jose Verissimo	CS		
Lucio Marcenaro	UNIGE		
Kyriakos Stefanidis	ATHENA		
Gordana Mijic	THYIA		
Klemen Selcan	THYIA		
Reviewed by			
Name	**Company**		
Iñaki Arenaza	MGEP		
Spase Drakul	THYIA		
Approved by			
Name	**Company**		

Modification History

Issue	Date	Description
Draft A	04 March 2011	First issue for comments (midterm progress report)
Issue 1	15 March 2011	Incorporates comments from Draft A review
Issue 1	31 October 2011	Final report. First complete draft.
Issue 2	11 November 2011	Final report. Final reviewed version.

Contents

Figures

Tables

Glossary

BER	Bit Error Rate
CPS	Cyber-Physical System
CR	Cognitive Radio
CRN	Cognitive Radio Network
DoS	Denial of Service
EBS	Exclusion Based System
ES	Embedded System
GPS	Global Positioning System
HIDS	Hybrid Intrusion Detection System
IC	Integrated Circuit
ICT	Information and Communication Technologies
IDS	Intrusion Detection System
LEAP	Lightweight Extensible Authentication Protocol
MAC	Message Authentication Code
PTX	Transmitter Power
RWG	Random-Walk Gossip
SCADA	Supervisory control and data acquisition
SDR	Software Defined Radio
SINAD	Signal-to-noise and distortion ratio
SNR	Signal to Noise Ratio
SPD	Security Privacy Dependability
TTL	Time To Live
WLAN	Wireless Local Area Network
WSAN	Wireless Sensor and Actuator Network
WSN	Wireless Sensor Network
XML	Extensible Mark-up Language

This Page is intentionally left blank

1 Trusted and Dependable Connectivity

This document together with deliverable *D4.1 SPD Network Technologies Prototype* complete the deliverables produced in WP4: SPD Network. Actually, D4.1 is contained in D4.2 which us the final deliverable of WP4. In addition to D.4.1, this document focus on SPD issues from a cognitive point of view with emphasis on the network functional layer of the pSHIELD system concept.

An important objective is the implementation of a radio system capable to maintain awareness of the operating scenario, to detect possible threats and to counteract in such a way to assure communications integrity to the maximum possible extent by reconfiguring the single nodes and/or the system itself. The set composed by awareness, threat detection, re-configurability, and reaction strategies forms what is named a cognitive radio. **Please refer to deliverable D4.1 for a detailed description of the Smart SPD driven transmission prototype.**

This document first describes the research related to the state-of-the-art technology within the means of providing security in lightweight and networked embedded devices through an adequate cryptographic scheme. After the research studies and evaluations performed in the first period, in the second period the main activities concerned preliminary studies and discussions regarding the setup of a general framework for secure communications within heterogeneous networks comprising resource-limited devices (pSHIELD application scenario). These activities included studies of SotA cryptography libraries available for resource constraint devices, a complementary study to test the respective algorithms implementation on the hardware of a possible micro (TelosB mote) and power (Linux based computer) node, and the selection and description of cryptographic libraries to be used.

Second, the study of the requirements for lightweight link-layer secure communication in wireless sensor network scenarios and the design and development of proper schemes focusing on confidentiality. More specifically, intrusion detection systems (IDS) have been studied. Misuse detection based IDS monitors the activities of a system and compares them with signatures of attacks that are stored in a database. This kind of IDS have high accuracy rates, however, due to the high increase of new attacks and the continuous variants of them it is extremely difficult to have an updated set of rules. On the other hand, anomaly detection depends greatly on the supposition that users and networks behave in a sufficiently regular way and therefore, any significant deviation from such behaviour could be considered as an evidence of an intrusion. Hybrid IDS, where the system is based in anomaly and misuse techniques best fit in WSN. However, there are application areas, such as SCADA systems, where anomaly detection performs better than in traditional information and communications technology (ICT) networks. SCADA communications are deterministic, and their operation model is often cyclical. Based on this premise, modelling normal behaviour by mining specific features sets gets feasible and efficient.

Another important issue is the architecture deployed for the IDS. Attacks can be detected locally in nodes, centralized in a main processing node or even through the collaboration of global and local agents integrated in the application layer of nodes. Although it may result in an increase in the resource requirements of a sensor node, the global security level that can be achived using distributed intrusion detection is considered more reliable than the centralized one. The centralized architecture could not detect as many attacks, due to the low data rate of wireless communication and energy constraints of sensor nodes that could not afford the transmission of massive audit data to a base station. However, in a distributed intrusion detection system, no node is trustful, due to potential inside attackers. For that reason it is necessary to propose an agent able to detect anomalies in its host neighbours. The protection of the nodes is also necessary so it is highly recommended to implement a local agents in the nodes thar are able to analyse possible local feature changes.

Other activities include the design of distributed self-management and self-coordination schemes for unmanaged and hybrid managed/unmanaged networks, aiming to reduce the vulnerability to attacks depleting communication resources and node energy have also been carried out. While Confidentiality, Data Integrity and Service Availability are also addressed by security systems in wired networks, Energy consumption is a unique characteristic to the wireless sensor networks due to the resource limitation constraint. Regarding energy there is a necessity to asses the existing protocols and applications in different real situations as they are initially designed and studied in a simulated environment. We have studied the resource footprint (energy consumption among them) and its impact on performance on some commercially available devices. We could see both how different aspects of the communications protocol contributed to the footprint and how this in turn affected the performance. The methodologies used can be applied to other protocols and applications, aiding in future optimisations. Vulnerabilities in the communications protocol could lead to greater energy consumption and eventually to a DoS attack.

According the WP4 objectives, new technologies enabling smart SPD driven transmissions have been proposed. In particular, the Cognitive Radio (CR) paradigm, which is usually based on Software Defined Radio (SDR), has been proposed to deal with such transmissions. CR is composable and expandable and modular by definition. In fact, it has been designed to accommodate these features. The implemented Cognitive Radio Node is able to receive radio parameters from moving hosts and automatically detect possible threats. The internal architecture of the Node learns typical safe environments features thus detecting the presence of external attackers by analysing radio parameters. In the considered scenario, the cognitive node always updates the radio parameters (SNR, BER and Transmitter Power, PTX) for the self-awareness purposes. The design and development of the Smart Transmission Layer in SHIELD will rely on waveform-agile implementations on Software Defined Radio (SDR) platform interconnected with personal computer. Joint and cooperating implementations of security procedures over several communication standards are expected to be accomplished and improve the state-oft-the-art (SotA). As well, expected benefits at this point are in flexibility to join different security procedures over a range of different communication standards, and in easy integration with hardware security components.

1.1 SDR definitions

First of all, it is useful to review the design of a conventional SDR. Figure 1-1 shows a block diagram of a generic digital radio, which consists of five sections:

- The **antenna** section, which receives (or transmits) information encoded in radio waves.
- The **RF front-end** section, which is responsible for transmitting/receiving radio frequency signals from the antenna and converting them to an intermediate frequency (IF).
- The **ADC/DAC** section, which performs analog-to-digital/digital-to-analog conversion.
- The **digital up-conversion** (DUC) and **digital down-conversion** (DDC) blocks, which essentially perform modulations of the signal on the transmitting path and demodulation of the signal on the receiving path.
- The **baseband** section, which performs operations such as connection setup, equalization, frequency hopping, coding/decoding, and correlation, while also implementing the link layer protocol.

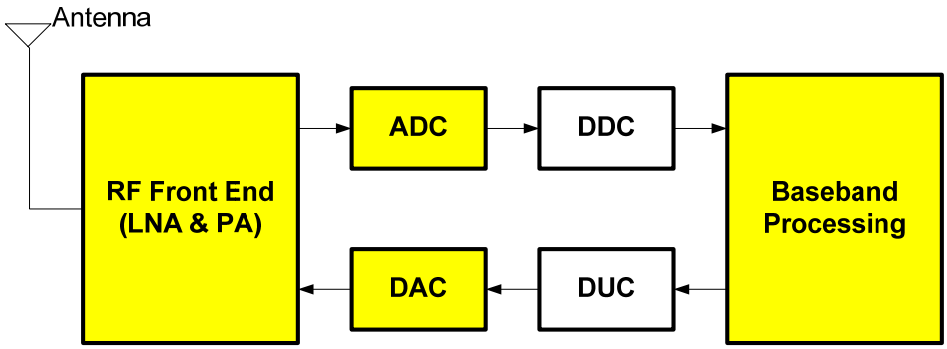

Figure 1-1: Schematic block diagram of a digital radio.

1.1.1 Embedded SDR solutions

Waveform processing can be performed on four different types of hardware platforms and configurations:

- General Purpose Processor (GPP)
- General Purpose Processor (GPP) + Digital Signal Processor (DSP)
- Field Programmable Gate Array (FPGA)
- Application Specific Integrated Circuit (ASIC)

While a large number of SDR products has been developed for running on a GPP (for example, in a desktop computer), the constraints of running on a EDs and the interest in using SDR on such devices have presented new challenges for SDRs. The user requirements include small size and limited weight, and long battery life. The challenge is to create SDR systems capable of meeting these constraints when running on embedded devices.

1.1.2 Comparison of embedded SDR solutions

The main advantage of SDRs is their flexibility. For this reason the best embedded solutions for such systems are the GPP, DSP, and FPGA. The main limitation of the first two of these systems is their performance. To increase their performance, a hybrid configuration can be created in which the GPP and DSP cooperate to achieve higher performance. In such a system the GPP controls the DSP and coordinates tasks, while implementing the most computationally demanding operations in the DSP. General purpose I/O operations are performed by the GPP. Although the global system performance is increased, the complexity of programming is increased since the programmer must deal with the communication between the two cores. Furthermore, since data must be sent over a communication channel, the potential parallelism may not be fully exploited. The GPP+DSP configuration, represents the main trend in the integration of SDRs in embedded devices. Nevertheless, FPGAs are used for performance critical tasks where the performance provided by the GPP+DSP system is insufficient. An example of this is the use of an FPGA in the Universal Software Radio Peripheral (USRP), where the FPGA is used for the signal decimation and for converting a signal to and from baseband14. Table 1-1 summarizes the comparisons made in this section. In this table the scores are from 1 (worst) to 5 (best) and they are related to each other.

Table 1-1: Comparison of embedded SDR solutions

Solutions	DSP	GPP	FPGA	ASIC	GPP + DSP
Flexibility	5	5	3	1	5
Performance	2	1	4	5	3
Programmability	4	5	2	1	4
Development cycle	5	5	3	1	5
Cost	5	4	3	1	4
Power consumption	2	2	4	5	1

1.1.3 Levels of SDR

It is not always feasible or practicable to develop a radio that incorporates all the features of a fully SDR. Some radios may only support a number of features associated with SDRs, whereas others may be fully software defined. In order to give a broad appreciation of the level at which a radio may sit, the SDR Forum (now called the Wireless Innovation Forum, WINNF) has defined a number of tiers. These tiers can be explained in terms of what is configurable.

- Tier 0: A non-configurable hardware radio, i.e. one that cannot be changed by software.
- Tier 1: A software controlled radio where limited functions are controllable. These may be power levels, interconnections, etc. but not mode or frequency.
- Tier 2: In this tier of software defined radio there is significant proportion of the radio is software configurable. Often the term software controlled radio, SCR may be used. There is software control of parameters including frequency, modulation and waveform generation / detection, wide/narrow band operation, security, etc. The RF front end still remains hardware based and non-reconfigurable.
- Tier 3: The ideal software radio or ISR where the boundary between configurable and non-configurable elements exists very close to the antenna, and the "front end" is configurable. It could be said to have full programmability. In Figure 1-1 it i depicted with yellow colour.
- Tier 4: The ultimate software radio or USR is a stage further on from the Ideal Software Radio, ISR. Not only does this form of software defined radio have full programmability, but it is also able to support a broad range of functions and frequencies at the same time. With many electronic items such as cellphones having many different radios and standards. A software definable multifunction phone would fall into this category.

Although these SDR tiers are not binding in any way, they give a way of broadly summarising the different levels of software defined radios that may exist.

1.1.4 Cognitive radio

The reconfigurability offered by SDR technology enables radios to switch functions and operations. However, an SDR can do this only on demand; it is not capable of reconfiguring itself into the most effective form without its user even knowing it. According to Mitola's early vision, a CR would be realized through the integration of model-based reasoning with software radio and would be trainable in a broad sense, instead of just programmable. The radio can reconfigure itself through an ongoing process of awareness (both of itself and the outside world), perception, reasoning, and decision making. The concept of CR emphasizes enhanced quality of information and experience for the user, with cognition and reconfiguration capabilities as a means to this end. Today, however, CR has become an all-encompassing term for a wide variety of technologies that enable radios to achieve various levels of self-configuration, and with an emphasis on different functionalities, ranging from ubiquitous wireless access, to automated radio resource optimization, to dynamic spectrum access for a future device-centric interference management, to the vision of an ideal CR. Haykin, for example, defines CR as a radio capable of being aware of its surroundings, learning, and adaptively changing its operating parameters in

real time with the objective of providing reliable anytime, anywhere, and spectrally efficient communication. The U.S. Federal Communications Commission (FCC) uses a narrower definition for this concept: "A Cognitive Radio (CR) is a radio that can change its transmitter parameters based on interaction with the environment in which it operates. The majority of cognitive radios will probably be SDR (Software Defined Radio) but neither having software nor being field programmable are requirements of a cognitive radio." Despite these differences in both the scope and the application focus of the CR concept, two main characteristics appear to be in common in most definitions. They are reconfigurability and intelligent adaptive behavior. Here by intelligent adaptive behavior we mean the ability to adapt without being a priori programmed to do this; that is, via some form of learning. For example, a handset that learns a radio frequency map in its surrounding could create a location-indexed RSSI vector (latitude, longitude, time, RF, RSSI) and uses a machine-learning algorithm to switch its frequency band as the user moves.

From this it follows that cognitive radio functionality requires at least the following capabilities:

- **Flexibility and agility**: the ability to change the waveform and other radio operational parameters on the fly. In contrast, there is a very limited extent that the current multi-channel multi-radio (MCMR) can do this. Full flexibility becomes possible when CRs are built on top of SDRs. Another important requirement to achieve flexibility, which is less discussed, is reconfigurable or wideband antenna technology.
- **Sensing:** the ability to observe and measure the state of the environment, including spectral occupancy. Sensing is necessary if the device is to change its operation based on its current knowledge of RF environment.
- **Learning and adaptability:** the ability to analyze sensory input, to recognize patterns, and modify internal operational behavior based on the analysis of a new situation, not only based on precoded algorithms but also as a result of a learning mechanism. In contrast, the IEEE 802.11 MAC layer allows a device to adapt its transmission activity to channel availability that it senses. But this is achieved by using a predefined listen-before-talk and exponential backoff algorithm instead of a cognitive cycle.

1.1.4.1 Different interpretations of SDR

Table 1-1 shows a comparison of different interpretations of CR. The most common aspects of all these interpretations is radio spectrum, as well as spectrum efficiency and primary users.

Table 1-2: Comparison of different interpretations of CR.

Aspects	Mitola	Haykin	SDR Forum	FCC	Inf. Theory
User's needs	x				
Context	x				
Intellig. & contr.	x	x	x		
Radio/spectr.	x	x	x	x	x
Spectr. effic.		x	x	x	x
Primary users		x	x	x	x
SDR	x	x			
Cooperation				x	
Reliability		x			

We also need to emphasize that there is yet another ambiguity in the definition of CN, since we cannot equate CN and cognitive radio network (CRN). For example, CN is defined as a network constructed of primary and secondary users, where secondary users are considered the cognitive ones. These users simply obtain the additional information on the activity of the primary users to employ better transmission parameters, in this context limited only to coding. Cognitive networks are wireless networks that consist of two types of users:

- **PRIMARY USERS:** These wireless devices are the primary license holders of the spectrum band of interest. In general, they have priority access to the spectrum and are subject to certain quality-of-service (QoS) constraints that must be guaranteed.

- **SECONDARY USERS**: These users may access the spectrum, which is licensed to the primary users. They are thus secondary users of the wireless spectrum and are often envisioned to be cognitive radios. For the rest of this chapter, we assume the secondary users are cognitive radios (and the primary users are not) and use the terms interchangeably. These cognitive users employ their "cognitive" abilities to communicate while ensuring the communication of primary users is kept at an acceptable level.

1.1.5 Types of adaptable radio devices

Table 1-3 summarizes some types of adaptable radio devices.

HARDWARE RADIO: The capability of CR devices changing their radio characteristics is implemented completely in hardware. Thus, once in the field the devices will not be able to change their characteristics other than what is already built in. For example, the range of frequency programmed into the hardware always remains the same, even though the user knows that there is an opportunity to work in a different range. Therefore, the scope is limited in this case.

SOFTWARE RADIO: The capability of CR devices changing their radio characteristics also is implemented in software. Thus, the devices are able to change their characteristics from other than what is already built in. For example, contrasting with the preceding, the range of frequency programmed into the hardware may be changed by uploading a newsoftware patch (say, a simple configuration file).

ADAPTIVE RADIO: This is the capability of CR devices where its radio characteristics are changed by mechanisms such as closed-loop or open-loop controllers. Basically, the devices adapt to the surroundings by sensing and using the preprogrammed logic and control techniques.

RECONFIGURABLE RADIO: The radios in CR devices of which the functionalities can be changed manually. A hardware radio and a software radio both are reconfigurable, though in different ways and to different degrees.

POLICY-BASED RADIO: The changes to the radio functionalities of CR devices are governed by the policies. The policy set usually is available as a data set (or database). For example, the frequencies used by military equipment are not allowed to be used by others under all circumstances. Basically the policy set governs the operational characteristics of the CR devices quite immaterial of whether they are capable.

COGNITIVE RADIO: It has been already defined. This includes databases, policies, learning techniques, and so forth.

INTELLIGENT RADIO: This includes cognitive radios, which are also able to learn as well as predict the situations and adapt themselves. In a general and crude sense, it is a software radio. However, with respect to the previous explanation of the software radio, it just specifies the capability to work with a software control, thus an intelligent radio is much more than a simple software radio.

Table 1-3: Types of adaptable radio devices.

Types of Radio	Platform	Reconfiguration	Intelligence
Hardware	HW	Minimal	None
Software	HW/SW	Automatic	Minimal
Adaptive	HW/SW	Automatic /predefined	Minimal/none
Reconfigurable	HW/SW	Manual/predefined	Minimal/none

Policy based	HW/SW	Manual (data based)/automatic	Minimal/none
Cognitive	HW/SW	Full	Artificial/machine laearning
Intelligent	HW/SW	Full	Machine learning /predicting decision

1.1.6 Standards

1.1.6.1 IEEE SCC41

The activities of IEEE SCC41 are supported by IEEE Communication Society technical committees, which represent the interface with the research community in terms of proposals for new standards and technical contributions to the internal discussions within the working groups (WGs).

1.1.6.1.1 IEEE 1900.1

IEEE decided to create 1900.1 WG, Standard Definitions and Concepts for Spectrum Management and Advanced Radio System Technologies, responsible for creating a glossary of important CR-oriented terms and concepts. It further provides explanations to germinate a coherent view of the various efforts taking place in the broad arena of CR. The key idea was to standardize and prepare technically precise definitions related to CR. In fact, 1900.1 WG acted as a glue to the other IEEE SCC41 WGs, tying them together with common definitions of CR terms. The IEEE 1900.1 has been voted by the IEEE Standard Association and is a standard now.

1.1.6.1.2 IEEE 1900.2

In light of new CR technology, many radio systems coexist and they try to optimize the utilization of spectrum in space and time. The accurate measurement of interference has thus become a crucial requirement for the deployment of these technologies. The mandate of the 1900.2 WG, Recommended Practice for Interference and Coexistence Analysis, was to recommend the interference analysis criteria and establish a well-thought-out framework for measuring and analyzing the interference between radio systems. New technologies, while attempting to improve spectral efficiency—by being flexible, collaborative, and adaptive—also cause disputes. The framework for interference analysis addresses the context of measurements and the purpose. Any new adaptive system has a trade-off between cost and gain. Therefore, the interference analysis should make this gain explicit, along with the usage model for this trade-off. Apart from the interference power measurements and the context, impact and remedies are also mentioned for analysis and comparison. Finally, parameters for analysis are derived from scenarios including the context and harmful interference thresholds. Uncertainty levels in measurements are compulsorily considered in the analysis. Just like 1900.1, the 1900.2 has been voted by IEEE Standard Association and has become a standard.

1.1.6.1.3 IEEE 1900.3

The aim of IEEE 1900.3 is to define a set of recommendations that helps in assuring the coexistence and compliance of the software modules of CR devices before proceeding toward validation and certification of the final devices, as laid down in IEEE 1900.2. Since SDR is an important component of future CR networks, these recommended practices should help in creating high confidence in the deployed SDR devices. These devices will have multiple layers of software, each addressing different functionalities. Therefore, it is all the more essential to test the capability of SDR devices a priori to install the patches correctly over the air, assuring secure execution of intended functionalities. As an illustration consider an implementation of the SDR device specifications into a program. This can be verified with the formal verification methods. However, formal specifications for software, mostly, do not exist. Therefore, testing in these cases becomes less formal, by focusing on only a particular subset of device operations. The aim is to design testing procedures that will comply with the semi-formal software specifications. One of the solutions is to define checkpoints (mandatory or obligatory) and assertions that will reflect the

specifications. For these reasons IEEE 1900.3 WG specifies device management procedures. Since many of those exist today (e.g., Java's Mobile Device Management Server application programming interfaces), 1900.3 WG utilizes other relevant standards to achieve its goal.

1.1.6.1.4 IEEE 1900.4

The 1900.4 WG, Coexistence Support for Reconfigurable, Heterogeneous Air Interfaces, defines the overall system architecture, splitting the functionality between terminals and the network, and the information exchange between coordinating entities. Its main goal is to increase the overall system utilization of reconfigurable terminals while increasing the perceived quality of service. All 1900.4-enabled devices should operate in an OSA or DSA manner so that they will not degrade the performance of PU radio access devices. The study of heterogeneity in wireless access technologies and multihoming of the devices—with CR capability—differentiates this WG from other WGs of SCC41. At first the 1900.4 WG looks into only the architectural and functional definitions. The corresponding protocol definitions related to the information exchange are addressed at a later stage. This standard was approved in late January 2009. After the successful work and much interest in this WG, two more projects have been assigned to 1900.4 WG:
1900.4a. Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks—Amendment: Architecture and Interfaces for Dynamic Spectrum Access Networks in White Space Frequency Bands.
1900.4.1. Standard for Interfaces and Protocols Enabling Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Networks.
With these two projects, the scope of 1900.4 has expanded IEEE's interest in the community.

1.1.6.1.5 IEEE 1900.5

The recent WG of IEEE SCC41, started in August 2008, on Policy Language and Policy Architectures for Managing Cognitive Radio for Dynamic Spectrum Access Applications, defines a policy language (or a set of policy languages or dialects) to specify interoperable, vendor-independent control of CR functionality and behaviour for DSA resources and services. The initial work concentrates on standardizing the features necessary for a policy language to be bound to one or more policy architectures to specify and orchestrate the functionality and behavior of CR features for DSA applications (see www.scc41.org/5).

1.1.6.1.6 IEEE 1900.6

Yet another WG of IEEE SCC41 started in August 2008, on Spectrum Sensing Interfaces and Data Structures for Dynamic Spectrum Access and Other Advanced Radio Communication Systems. The intended standard defines the information exchange between spectrum sensors and their clients in radio communication systems. The logical interface and supporting data structures used for information exchange are defined abstractly without constraining the sensing technology, client design, or data link between sensors and clients.

1.1.6.2 IEEE 802.22 for TV White Space

IEEE 802.22 is thought of as an alternative technology to WiFi with an unlicensed spectrum like that ofWiFi, but a better spectrum between 54 MHz and 863 MHz. Similar to TV signals the access to Internet could be over tens of kilometers and no restrictions regarding in-building environments and the like. However, the challenges were many: identification of the primary users, listing the unused channels locally, and defining the power levels so as not to interfere with the adjacent bands. The two important entities defined here are the base station and customer premises equipment (CPE). BS controls all the CPEs, determining when to send data and the channels to use. CPEs also sense the spectrum in its vicinity, enabling distributed sensing, and send it back to the BS. With the opening up of TV white space by the FCC, this standard gained a significant role.

1.1.6.2.1 TV White Space

Digital Switchover (DSO) was completed in the US in June 2009, and is expected to be completed in the UK by 2012. A similar switchover process is also underway or being planned (or is already completed) in the rest of the EU and many other countries around the world. After Digital Switchover a portion of TV analogue channels become entirely vacant due to the higher spectrum efficiency of digital TV (DTV). These cleared channels will then be reallocated by regulators to other services through auctions.

1.1.6.3 IEEE 802.11af White-Fi

White-fi is a term being used to describe the use of a Wi-Fi technology within the TV unused spectrum, or TV white space. The IEEE 802.11af working group has been set up to define a standard to implement this. With a number of administrations around the globe taking a more flexible approach to spectrum allocations, the idea of low power systems that are able to work within portions of spectrum that may need to be kept clear of high power transmitters to ensure coverage areas do not overlap is being seriously investigated. When using systems like white-fi, IEEE 802.11af that use TV white space, the overall system must not cause interference to the primary users. With processing technology developing further, this is now becoming more of a possibility. In order for white-fi 802.11af to be able to operate, it is necessary to ensure that the system does not create any undue interference with existing television transmissions. To achieve this there are a number of technologies and rules that may be utilised. One way in which a white-fi system would be able to operate is to use cognitive radio technology.

1.1.6.4 ETSI's Reconfigurable Radio Systems Technical Committee

The ETSI Technical Committee (TC) on Reconfigurable Radio Systems (RRS) has the responsibility for standardization activities related to Reconfigurable Radio Systems encompassing system solutions related to Software Defined Radio (SDR) and Cognitive Radio (CR), to collect and define the related Reconfigurable Radio Systems requirements from relevant stakeholders and to identify gaps, where existing ETSI standards do not fulfil the requirements, and suggest further standardization activities to fill those gaps.

1.1.6.5 ITU-R

The studies are being conducted within International Telecommunication Union – Radio communication Sector (ITU-R) Working Party 1B, with input from other Working Parties describing software defined and cognitive radio applications in their areas of competence. The studies are broadly focused, with ITU-R merely invited "to study whether there is a need for regulatory measures related to the application of" cognitive technologies or software defined radio. With a focus this broad, the results can be wide-ranging, and provide as much opportunity for mischief as for a helpful outcome. Much of the work so far has been focused on potential definitions for software defined and cognitive radio systems. The concept and definition of software defined radio is proving much less controversial than those of cognitive radio systems, largely because cognitive systems don't easily fit into the existing paradigm of spectrum management via allocations to services. Although revolutionary in many ways, software defined radio functions in the same practical way that a conventional radio does when transmitting and receiving. They are easily used within the existing regulatory framework, and no change is likely to the Radio Regulations beyond a definition.

1.1.7 Taxonomy of attacks on SDR

SDR attack taxonomy is defined as:

1. Interception (Confidentiality)
   - SW piracy
   - Loss of anonymity
   - Private configuration exposure
2. Modification (Integrity)
   - Unit malfunction

- Change of preference
- Security function circumvention

3. Interruption (Availability)
   - Jamming
   - Malicious code
   - Resource exhaustion
4. Fabrication (Authenticity)
   - Rogue terminal
   - Financial fraud
   - Network impersonation

CRN security threats are defined as:

A. Spectrum access – related security threats
   1. Threats to incumbent coexisting mechanisms
      - Spectral “honeypots”
      - Sensory manipulation:
        - Primary-user emulation
        - Geospatial manipulation
        - Tx false spectrum sensing information
      - Obstruct synchronisation
   2. Threats to self-coexistence mechanisms
      - Tx false/spurious intercell, beacons (control messages)
      - Exploit/obstruct intercell, spectrum sharing process
B. Radio SW security threats
   - Security threats to the SW download process
   - Injection of false/forget polices
   - Injection of false/forgot SW updates
   - Injection of malicious SW (viruses)
   - SW IP Theft
   - Software tempering
     - Unauthorised policy changes
     - Tempering w/ CR reasoners (e.g., system strategy reasoner & policy reasoner)

1.1.7.1 Attacks to CRNs

Security is necessary in CRNs because the data channel is easily accessed by an attacker. In the context of CRNs, we define attacks as actions that achieve at least one of the following goals:

- **Unacceptable interference to licensed primary users:** Because of the attack, the communication channel of primary/licensed users of a frequency band is diminished or just becomes unusable (denial-of-service (DoS) attack).
- **Missed opportunities for secondary users:** An attacker could prevent secondary users from using available spectrum bands thus, once again, reducing channel performance or just denying service to secondary users.
- **Access to private data:** An attacker could try to access data in an unauthorized way. As a consequence data must be secured by means of cryptographic primitives.
- **Modification of data:** An attacker could try to modify the data exchanged between several entities to its own advantage. Thus, integrity of data must be assured.
- **Injection of false data:** Injection of false data could make the CRN to perform in an unpredictable way or just following the attacker guidelines. Therefore, authentication of information sources should be guaranteed.

In CRNs emerges the possibility of new specific attacks, which are:

- PUEA (primary user emulation attacks),
- OFA (objective function attacks),
- CCDA (common control data attacks),
- False feedback, and
- Lion attack.

1.1.7.1.1 PUEA

Primary user emulation attack (PUEA) is first identified by Chen and Park in 2006. In PUEA, an attacker occupies the unused channels by emitting a signal with similar form as the primary user's signal so as to deter the access of the vacant channels from other secondary users. The cognitive radios have highly reconfigurable air interface which makes it possible for an attacker to modify the air interface to mimic a primary user signal's features and thereby leading legitimate secondary users to erroneously identify the attacker as a primary user. The investigation shows that a PUE attacker can severely compromise the spectrum sensing performance and significantly reduce the channel availability to legitimate secondary users. PUEA can compromise a cognitive radio system using either of spectrum sensing methods to attack the energy detection scheme, PUE attacker may masquerade the primary user by transmitting signal with the similar energy as primary user; to defeat cyclostationary detectors, an attacker can make its transmissions indistinguishable from primary user signals by transmitting signals that have the same cyclic spectral characteristics as primary user signals. The fundamentals of PUEA is that the adversary is not focus on jamming primary users, but on forestalling idle spectrum bands that could have been used by other secondary users. Depending on the motivation behind the attack, a PUE attack can be classified as either a selfish PUE attack or a malicious PUE attack. A selfish PUE attacker aims to prevent other secondary users from competing for that band by sending signals with similar characteristics of primary user signals whereas a malicious user launching an attack in the same manner, is more interested in obstructing the whole dynamic spectrum access process rather than monopolizing the utilization of the frequency spectrum resource. Depending on the motivation behind the attack, a PUE attack can be classified as either a selfish PUE attack or a malicious PUE attack:

- **Selfish PUE attacks**: an attacker's objective is to maximize its own spectrum usage. When selfish PUE attackers detect a fallow spectrum band, they prevent other secondary users from competing for that band by transmitting signals that emulate the signal characteristics of primary-user signals. This attack is most likely to be carried out by two selfish secondary users of which the intention is to establish a dedicated link.
- **Malicious PUE attacks**: the objective of this attack is to obstruct the DSA process of legitimate secondary users; that is, prevent legitimate secondary users from detecting and using fallow licensed spectrum bands, causing denial of service. Unlike a selfish attacker, a malicious attacker does not necessarily use fallow spectrum bands for its own communication purposes. It is quite possible for an attacker to obstruct the DSA process simultaneously in multiple bands by exploiting two DSA mechanisms implemented in every CR. The first mechanism requires a CR to wait for a certain amount of time before transmitting in the identified fallow band to make sure that the band is indeed unoccupied. Existing research shows that this time delay is no negligible. The second mechanism requires a CR to periodically sense the current operating band to detect primary-user signals and immediately switch to another band when such signals are detected. By launching a PUE attack in multiple bands in a round-robin fashion, an attacker can effectively limit the legitimate secondary users from identifying and using fallow spectrum bands.

1.1.7.1.2 OFA

Within a CRN, incumbents control several radio parameters to enhance the network performance. The parameters choice is often done by means of an artificial intelligence (AI) algorithm, such as genetic, hill-climbing, or random walks. Such algorithms make slight modifications of several input factors to find their optimal values that maximize an objective or goal function. In the context of CR, input factors can be frequency, bandwidth, power, modulation type, coding rate, channel access protocol, encryption type, authentication type, message integrity code, and frame size. It is necessary to remark that the OFA performance is much related to the amount of on-line learning of the CRN. The on-line learning refers to an on-line optimization of the search space. On the other hand, radios that perform off-line learning

observe the environment just once, and then search an optimal configuration off-line (e.g. just following a predefined radio policy); as the configuration of such radios are independent of their observations, off-line learning is not affected by OFAs. However, radio devices using only off-line learning do not theoretically require a learning engine and thus cannot be considered as CRs. So, every CRN is exposed to OFA attacks.

1.1.7.1.3 CCDA

In some approaches, a dedicated channel is used to exchange sensing information:

- between the base station and the secondary users if the CRN is centralized (i.e. DIMSUMnet) and
- between secondary users if it is distributed (such KNOWS or CORVUS)

A malicious user could jam this channel, disrupting all transmissions and preventing elements within the CRN from sharing information about spectrum usage. The lack of knowledge about available bands keeps the CRN from operating (DoS attack). Moreover, eavesdropping on the control data provides the attacker with all the required information to detect which new channel the CRN is switching to. The need of securing the common control data is hence patently obvious. 802.22 Working Group is aware of this threat and has proposed mechanisms to protect such information. We consider that the impact of this attack is more relevant in centralized CRNs as an attacker can focus on jamming the control channel within the base station vicinity (single point of failure) and thus easily affecting the whole network.

1.1.7.1.4 False feedback

Within a cooperative framework where secondary users exchange sensing information, false feedback from one or a group of malicious users could lead the CRN to take improper actions. For example, the CRN could conclude that a given frequency band is occupied by a primary user when actually it is not the case or, the other way round, it could consider it as a vacant band when being used by a primary network. In the former case, the attacker prevents the CRN from using an available band. In the latter, if the CRN decides to use that band to operate, transmissions of secondary users could harmfully interfere with primary signals. This risk is especially relevant for fully distributed CRNs because false feedback could be propagated, thus affecting a large portion of the network. Such an effect is often referred as a virus due to its undesired distribution, but opposite to the 'traditional' virus, it applies to the link layer instead of the application one. On the other hand, in centralized networks like 802.22 the station collects sensing measurements from all CRs to determine which frequency bands are occupied. Although the IEEE 802.22 standard establishes that the final decision on the availability of a channel must be performed at the base station, it does not specify how it must be made. Generally speaking, in this situation a malicious user could be easily detected, as the information provided by the latter may be incongruous.

1.1.7.1.5 Lion attack

The Lion attack is defined as a jamming targeted to reduce the throughput of TCP by forcing frequency handoffs. The handoff process involves sensing the medium looking for vacant channels and choosing the best one according to some criteria, thus incurring high latencies until the transmission is resumed. A malicious user trying to disrupt a TCP connection of a secondary user can perform a PUEA to force a handoff in the CRN. As the transport layer is not aware of the disconnection, it keeps sending data segments which are queued at lower layers but not transmitted and thus TCP segments can be delayed or even lost. As the TCP sender is allowed to transmit new data upon reception of acknowledgments, loss or delay of segments can lead to a period of inactivity of the former.

2 Security in Embedded Systems

The modern day embedded systems (ES) employ increasingly sophisticated communication technologies: low-end systems, such as wireless head-sets use standardised communication protocols to transmit data, remotely-controlled thermostats adjust room temperatures on user request sent from a mobile phone or from the Internet, while smart energy meters automatically communicate with utility providers. Furthermore, wireless sensor networks (WSN), or the recently emerging cyber-physical systems (CPS) are proposed to autonomously monitor and control safety-critical infrastructure such as, for example, a nation-wide power grid [1]. The increased complexity of these systems and their exposure to a wide range of potential attacks involving their communication interfaces makes security an extremely important and, at the same time, challenging problem.

The pSHIELD project recognizes the fact that security, privacy and dependability (SPD) are core characteristics of any modern ES and it proposes to address them as a "built-in" technology rather than as "add-ons". In fact, due to the complexity of networked embedded systems, as well as because of the potentially high cost of failures, SPD must become an integral part of ES design and development [2].

2.1 Networked Embedded Systems

The current trends in ES design show a strong tendency towards the use of wireless communications, as well as of small, low-cost devices with sensing capabilities. The process started with the spread of mobile communications and, later, accelerated with the proliferation of local area wireless communication technologies such as Wireless LAN or Bluetooth. More recently, the development of low-cost, integrated wireless transceivers and MEMS sensors resulted in an explosion of research in the new field of wireless sensor networks.

Currently, wireless sensor networks are gradually making their way to the market promising near real-time monitoring of potentially large-scale areas [3]. Recent research proposes to extend distributed monitoring with actuation enabling this way distributed control of spatial processes and leading to a multitude of new applications that range from large-scale fire-prevention systems, through automated building energy management, to large-scale control of industrial systems and infrastructures. These technologies are often referred to as cyber-physical systems or wireless sensor-actuator networks (WSANs) and, although still in their infancy, they are widely expected to become dominant market drivers in the coming [4].

These trends are further strengthened by the on-going standardization of wireless communication protocols for industrial applications such as, for example, Zigbee [5], ISA [6] or WirelessHART [7] and it is, therefore, reasonable to assume that the future embedded systems are likely to have at least some of the following characteristics:

- **Resource constraints:** Small, battery-operated wireless devices enable cheap sensing in hard-to-reach places and in harsh environments. The small-size factor and lack of cabling further increase the range of their possible applications in areas such as, for example, home appliances and consumer electronics. The advantages come, however, at a price of increased difficulty of software development and of securing the system due to the resource limitations, which usually take the form of small memory, low processing power and limited battery capacity.

- **Mobility:** Although fully autonomous systems may comprise only physically static devices, mobile network nodes might need to be used in many applications that require human interaction or supervision. These could take the form of personal data assistants (PDAs) or laptops and their presence adds to the complexity of securing the system since they may join and leave the network in different places in an unpredictable manner.
- **Heterogeneity:** The future embedded systems are likely to comprise devices of many different types. For example, a large scale monitoring and surveillance system could comprise different types of sensors such as, for example, digital cameras and passive infra-red (PIR) sensors, as well as data processing nodes and various actuators (e.g., remotely controlled door locks, sprinklers or alarms). Furthermore, industrial-grade distributed embedded systems might also require fixed infrastructure in the form of network routers, gateways and base stations. Security for heterogeneous embedded systems is challenging due to the fact that different parts of the system might have different computational capabilities, as well as different security requirements, thus precluding uniform application of the same security measures and techniques across the entire system.
- **Hierarchy:** Heterogeneous networked ES, especially when they comprise devices of radically different capabilities, often follow the hierarchical design pattern in which less capable devices are dependent on more powerful devices. This approach is a standard engineering practice in industrial control systems and has been recently suggested favourable for large-scale WSNs and WSANs in order to improve their overall energy efficiency and reliability [8].
- **Timeliness requirements:** Networked embedded systems that perform control tasks typically operate in a tight time regime, meaning that they need to execute control commands on time. Although the required degree of timeliness depends on the application, real time plays an important role in many ES and securing against timing-related attacks may prove difficult, as well as it is currently an active research topic [9].

All of these characteristics apply to the dependable surveillance system for urban railways, as described in the pSHIELD project's main application scenario. The system is envisioned to be a hierarchically-organised heterogeneous network of devices whose size and capabilities would span from large control room servers to small, battery-powered sensors.

2.2 Security Threats and Models

Networked embedded systems are envisioned to perform tasks upon which human safety and prosperity might depend. For example, failures (either random or inflicted by an attacker) of a railway infrastructure-monitoring system might put the lives of train passengers in danger while flaws in the security of a distributed surveillance system might lead to noticeable financial losses. However, securing networked, heterogeneous embedded systems with potentially constrained resources is a challenging task. A distributed embedded system might have many users and complicated usage patterns resulting in sophisticated access control policies. Wireless communications, as well as physical distribution of system's components across potentially large areas significantly increase the diversity of possible attacks the system is exposed to. Finally, the constrained resources of some of the system's components put serious limitations on the range of the available cryptographic primitives that can be used to secure it.

2.2.1 Attacks on Embedded Systems

There is a wide range of attacks that can be launched against embedded systems. The traditional Dolev-Yao [10] threat model focuses on the security of communication between two parties, in which each of which is considered to be secure and trusted (as a device). The model assumes that the attacker is able to overhear, intercept, capture and introduce its own messages to the communication

channel and it is up to the communication protocol to ensure confidentiality, integrity and authenticity of the transmitted messages. However, although general and applicable to a large class of communication systems, the model is not well suited to embedded systems because the physical exposure of embedded devices to potential manipulation renders them untrusted.

2.2.1.1 Attacks on Cryptosystems

There are a number of techniques that have been used in the past to exploit weaknesses of some cryptographic algorithms and are currently used as basic evaluation criteria for new algorithms. The common aim of these attacks is to reveal partially or entirely the information encrypted in intercepted messages, or to extract some information internal to the encryption process (without initially knowing any secrets). They include:

- **Brute force attack:** traversing the entire encryption key space in order to learn the encryption key.
- **Dictionary attack:** related to the brute force attack in that a set of keywords are used as possible values of the encryption key (or a pass phrase).
- **Chosen cypher text attack:** obtaining information about a secret decryption key by submitting a range of cipher texts to decrypt. .
- **Adaptive chosen cypher text attack:** a version of chosen cypher text attack in which the attacker interactively selects subsequent cypher texts based on the results of decryption of the previous ones.
- **Cypher text-only attack:** the attacker has access to a limited set of cypher texts.
- **Known plain text attack:** the attacker has access to a number of cypher texts together with the corresponding plain texts.
- **Chosen plain text attack:** the attacker can encrypt an arbitrary set of chosen plain texts.
- **Adaptive chosen plain text attack:** like above, but the attacker chooses subsequent plain text for encryption based on the previous results.
- **Related-key attack:** the attacker has access to encryption of a plain text under several different keys whose exact values may not be known but which are somehow mathematically related.

In addition to these general attack methods, there is also a range of more general cryptanalytic techniques that may be used to study the properties of cyphers. They include frequency analysis, differential cryptanalysis, linear cryptanalysis, statistical cryptanalysis and mod-n cryptanalysis. Finally, there are also attacks on hashing functions (e.g., birthday attack) that aim at finding collisions in hash functions, or attacks on random number generators that exploit a generator's statistical weaknesses to simplify breaking a cipher that uses it.

2.2.1.2 Attacks on Protocols

Communication and security protocols can be attacked in a number of ways by intercepting and inserting messages in the communication channel. These attacks are even easier to perform in wireless networks since there might be little difficulty in accessing the channel, unless a more sophisticated technology such as direct-sequence spread spectrum (DSSS) or frequency hopping are used.

- **Replay attack:** resending of some captured messages in order to confuse the protocol or to exploit some of its weaknesses.

- **Wormhole attack:** a form of a replay attack that uses a low-latency and long-range transmission link to intercept communications in one part of the network and then to reproduce them in another network region, for example, with the goal of authenticating the attacker.
- **Man-in-the-middle attack:** the attacker intercepts all communications from a node A, modifies them and sends to a node B in such a way that both A and B have the illusion of direct communication with each other.
- **Bit flipping attack:** selectively flipping bits in intercepted messages in order to achieve desired protocol behaviour, for example, to route traffic to different recipients or to change the message type.
- **Attack on key distribution protocols:** preventing or intercepting key distribution in the network might severely affect the entire safety infrastructure of the system.
- **Routing protocol attacks:** the attacker may influence the contents of routing tables of some network nodes or even to introduce corrupt nodes to affect communication in the network.

2.2.1.3 Denial of Service

The main task of all embedded systems is to interact with the environment they are embedded in. Thus, there is a shift in the goals a potential attacker might want to achieve from simply trying to steal or forge confidential information, to also trying to prevent the system from achieving its design goals or even to deliberately damaging it. The denial of service (DoS) attacks may include the following:

- **Physical damage**.
- **Jamming of communication lines:** particularly important when wireless communications are employed.
- **System overloading:** the attacker may send a large number of requests making the system incapable of normal operation.
- **Attacks on the system's power lines**.
- **Battery depletion attacks:** the attacker may disrupt the operation of communication protocols with the goal of using up the remaining energy of battery-powered devices. For example, wireless sensor nodes are typically battery-powered and wireless transmissions consume a significant amount of energy. Engaging a node in continuous communications will quickly drain its batteries. Also, the attacker might try to circumvent the operation of a duty-cycling protocol in order to increase the network's duty cycles.

2.2.1.4 Physical and Side-Channel Attacks

Many modern cryptographic protocols are designed in such way that their security depends on the key rather than on the secrecy of the protocol's design. Thus, the security of an ES can be circumvented if the attacker has physical access to some of the system's components and is capable of extracting the keys. Depending on the capabilities of ES hardware and on the attacker's resources, there are many types of side-channel attacks that can be realised and they can be generally grouped in two categories: invasive and non-invasive. The former refers to the attacks that require physical tampering with a device, for example, micro-probing and reverse design engineering, while the latter comprises attacks that aim at extraction of cryptographic secrets through the analysis of the external effects of a device's operation.

Typically, tamper-proof hardware technologies are used in order to defend against invasive attacks. The main idea behind them is to be able to detect abnormal usage situations by means of specialized sensors or circuits and then to destroy the sensitive parts of the system, for example, by zeroing the key

storage memory. Other techniques may include sealing for tamper evidence, tamper-proof casing, encryption of communication lines between hardware components (e.g., between the processor and the memory) or detection of abnormal clock rates and voltages.

The Federal Information Processing Standard FIPS 140-2 [11] defines four levels of physical security requirements for secure embedded systems: Level 1 devices have minimum physical protection, Level 2 devices implement tamper evidence mechanisms such as seals or enclosures, Level 3 devices implement both tamper evidence and tamper response mechanisms (e.g., sensitive memory zeroing) and Level 4 devices must implement all the requirements for Level 3 devices, as well as they must provide for environmental failure protection and response mechanisms. Thus, it is clear that there is a spectrum of options available for physical ES security and, typically, the desired security level is traded off against the financial cost of implementing it.

On the other hand, non-invasive side-channel attacks do not require devices to be physically accessed, thus they are usually easier and cheaper to perform. They include the following techniques:

- **Timing analysis:** the attacker might be able to infer the keys of an encryption algorithm by measuring small variations in the time the device needs to perform cryptographic computations.
- **Power analysis:** similarly to the timing attack, power analysis aims at the extraction of the device's secret information by precise measurements of the variations in the devices power consumption (the current draw) throughout the execution of its cryptographic algorithm.
- **Electromagnetic analysis:** extraction of security information through the analysis of the device's electromagnetic radiation. For example, [12] demonstrate that it is possible to reliably read the sequence of keystrokes from a neighbouring room by using a high-grade antenna.

2.2.1.5 Attacks on Control Systems

Control systems are a category of embedded systems that operate in a close connection with a real-world process such as, for example, a flight control system or a chemical plant controller. Usually, these systems make their control decisions in real time and might be vulnerable to timeliness inaccuracies the effects of which may sometimes be catastrophic. Following [9], the possible attacks might include:

- **Time synchronization attack:** the attacker can influence a distributed time synchronization protocol in order to desynchronize different parts of the system. Also, hardware manipulation or system overloading might be used to achieve the same effect.
- **Attack on sensors:** the attacker may influence responses of the system's sensors thus destabilising its control loop.

2.2.2 Attacker Model

Security engineering is a discipline in which one seeks to provide measures of dealing with the unexpected and security systems can only protect against an a priori known set of threats. Thus, in order to define the scope of the system, one needs to define the broadest possible set of attacks the system might be exposed to.

The most typical assumptions in the security literature correspond to the security of the communication links. We should assume, therefore, the following capacity of the attacker with respect to the messages transmitted by network nodes:

- Reception.

- Deletion.
- Insertion.
- Reordering.
- Corruption.
- Delay.

These assumptions encompass a wide range of possible attacks: deletion of messages can be used to model jamming of communication lines in wireless networks, while safety of such network protocols as key distribution, routing and duty cycling necessarily depends on the attacker's ability to modify message contents in an authentic way.

Another group of assumptions, crucial for the domain of ES, is the ability of the attacker to tamper with the network nodes. These, however, depend on the hardware technologies employed and on the expected resources of the attacker. There is already a large body of research on tamper-proof hardware and there is also a range of commercial products available (for an overview see [13]). Because none of these technologies offer 100 per cent security, their choice should necessarily involve matching their strength against their cost, the required security level and the expected resources of a potential attacker. For example, a wireless network of low-cost temperature sensors used in home environment could, perhaps, use less sophisticated tamper-proof technology, but a system responsible for surveillance of critical infrastructure should use the best technology available since the potential gain from corrupting it might attract attackers with substantial resources.

Since the tamper-proof security levels may vary between different embedded systems and they change over time (upgrades are needed as new types of attacks are devised), it is reasonable to assume that the attacker can extract secret information from all types of devices, but with a varying probability of success. As a result, for every type of technology used, an estimate can be made on the time needed for an expected attacker to compromise the device. This approach makes the provision of system's security more realistic by enforcing distribution of security responsibilities among other system components.

Finally, it has to be assumed that the attacker is able to command the compromised devices towards his goals and, thus, it has the ability to damage the system from the inside, unless proper counter measures (such as, for example, intrusion detection and separation of privileges) are implemented.

2.3 Security Requirements

This section discusses what should be expected from the security infrastructure in order to build a dependable ES. The basic requirements of any security system are typically concerned with the security of communication links and thus must include the following:

- **Authentication:** the ability to assure the identities of parties participating in a protocol. In the context of embedded systems, authentication typically means the ability to distinguish between the original and forged data packets and devices.
- **Integrity:** the ability to state whether communication messages have been tampered with. Integrity and authentication allow assuring both the origin and the contents of messages.
- **Confidentiality:** the ability to conceal the contents of communication messages. Confidentiality is, in general, orthogonal to authentication and integrity although encryption can be used to implement integrity.

These three basic requirements need to be extended in order to satisfy the threats ES can be exposed to. One of the key requirements for the security and dependability of an ES is graceful degradation of its services under random or inflicted (by an attacker) faults. A networked ES that comprises many nodes should gradually degrade the quality of its outputs as subsequent devices are either taken under the attacker's control or simply fail. This means, in particular, that after capturing one or few devices, the attacker cannot obtain access to the rest of the network, for example, by extracting the global network encryption key or by commanding the compromised devices to bring the network down. Instead, the damage has to be local, i.e., only the compromised devices and, perhaps, some of their network neighbours should be considered faulty. Also, as a consequence of the ability of the attacker to eventually compromise any device in the system, it has to be assumed that none of the network nodes can be fully trusted and measures need to be taken to ensure detection of abnormal node behaviour (an intrusion detection system should be in place).

If wireless sensor networks are to be a part of the system, then WSN-specific security requirements should be considered. In particular, the SPD security infrastructure should not preclude the ability of the sensor nodes to listen to their neighbours' traffic, to process data packets while they are routed towards their destination, as well as it should not preclude duty cycling.

Finally, we consider the following issues to be relevant to networked ES and, thus, require counter-measures to be implemented by the SPD middleware:

- Time synchronization is likely to be an integral part of any networked ES, thus it has to be implemented in a secure way.
- The capacity of the attacker to delete arbitrary messages already encompasses a DoS attack that is based on jamming communication lines. It is not sufficient, however, to simply conclude that an encryption or an authentication protocol should be able to handle such situation because this would render the system useless and unable to perform its main objectives. Therefore, robust wireless communications together with such techniques as channel and node black-listing should be implemented.
- Denial-of-service attacks that aim at battery depletion and system overloading should be prevented.
- The routing protocol used should be robust to maliciously-behaving compromised nodes.

2.4 Design Challenges

Meeting the presented set of requirements is a challenging task due to a number of reasons of which limitations in hardware capabilities and the available bandwidth play a central role. According to [2], there is a gap between the requirements of the available security protocols and the capabilities of the existing ES architectures. High grade cryptography exceeds the capabilities of small embedded processors and remains costly when applied to high-rate traffic on high-end servers. Furthermore, energy consumption overheads of security technologies are significant and, in the case of small battery-powered devices, they may become prohibitive.

Tamper-resistance technologies are absolutely necessary for ES applications because these systems often operate unattended and are thus exposed to a greater range of security risks than home or office PCs. In fact, since system's security is measured by the security of its weakest element, over-investing in cryptographic protocols and algorithms without due attention to its physical exposures might prove pointless. However, striking a perfect balance between the sufficient level of physical security and the cost of the system might be very difficult and it requires careful definition of the system's security goals.

Another issue is the difficulty of implementing security updates on networked embedded systems. Security systems are never perfect and updates are often required due to implementation faults, design flaws or new types of attacks being discovered. Updating an ES might prove difficult for a number of reasons. The system might comprise numerous devices, already deployed in places that potentially are hard to access, or it might be infeasible to switch off the system due to its importance. Finally, implementing a remote update functionality adds to its complexity and it might be a great source of vulnerabilities on its own.

Finally, security assurance, i.e., the probability that the system's security goals have been met, might be difficult to assert, especially in the case of real-time networked ES since their complexity and the range of security exposures grows significantly. Theoretical properties of security protocols matter only as long as their implementations are faithful. This issue is also closely related to the problem of security updates.

3 Cryptography Framework

3.1 Key Management

A key management scheme is an integral part of any deployed security system. Whether the cryptographic approach followed is symmetric or asymmetric, the role of an efficient key management scheme is vital. Such a scheme is affected by the system's architecture, device classes, deployment environment, potential attacks and other factors. For example, a key management scheme for a secure WSN needs to deal with the limitations of such a system in terms of nodes computational, storage and energy constraints in addition to expensive wireless communication. Essentially, key management comprises key pre-distribution approaches and other schemes dependent on the nature of the network. In all cases, certain basic operations should be supported such as key addition, revocation, and renewal. The organisation of this section is based on [93] and is in the light of applicability to sensor networks.

3.1.1 Conventional Schemes

Conventional key management schemes are normally based on the place where the keys are stored and their usability scope. Those are typically divided into pairwise, network-wide and centralized schemes.

3.1.1.1 Pairwise Key

A pairwise key scheme allows for node-to-node communication where each node shares a unique key with every other node in the network. Where such a scheme can provide high resilience against node capture and replication attacks it has, however, scalability issue. For example, if a network of 5000 nodes is to be deployed then every node needs to store 4999 keys in order to securely communicate with all other nodes. This is considered a prohibitive overhead especially in resource constrained devices.

3.1.1.2 Single Network-wide Key

As the name implies, a single key is used to encrypt and decrypt all communications among all network constituents. This is by far the simplest approach where before deployment all nodes are loaded with the same key. The advantages of such a scheme is in avoiding complex protocols and saving memory where there will be also no need for the key discovery phase. However, this scheme is highly vulnerable where if only one node is compromised, the whole network is immediately exposed.

3.1.1.3 Centralized/Trusted Base Station

In order to avoid exhausting storage resources at often constrained nodes, a centralised approach can be followed for key generation and distribution. Each pair of nodes in order to communicate need to receive a session key from a centralized server that has to be of utmost trustworthiness. This approach is resilient against node capture; however, it could face scalability issues as the number of nodes explode.

3.1.2 Key Pre-Distribution

Key pre-distribution is the scheme where keys are installed a priori in certain system entities, (e.g., sensors) using some specific distribution manner followed by a discovery process where pairs of entities determine the existence of a shared key for secure communication. A key pre-distribution scheme should guarantee, to a certain probability, that any two entities can communicate using a shared key.

A few key pre-distribution schemes exist. Some are mainly based on probabilistic methods that are sometimes enhanced through domain/application dependent information. Others use different sorts of heuristics.

3.1.2.1 Basic Random

In this scheme [86], the system follows three main phases; a pre-deployment phase, a shared-key discovery phase and paths establishment phase:

- In the pre-deployment phase, each node is loaded with K keys (key ring) randomly selected (without replacement) from a generated pool of keys. For each node, the key identifiers of a key ring and the associated node identifier are stored on a trusted controller node and the shared key is loaded.
- The key discovery phase is concerned with the initialization after the system's physical deployment where nodes try to discover other nodes sharing a common key identifier. Matching key identifiers from two separate key rings for two neighbouring nodes implies that the key of that identifier can be used as the shared key for secure communication. This step has to be done carefully in order to avoid traffic analysis attacks for example in the case of broadcasting key identifiers.
- The paths establishment phase is concerned with creating secure communication paths between nodes sharing a common key where some nodes can also act as mediators between other nodes. Mediation occurs when a node generates a pairwise key based on two other keys it shares separately with two other nodes which cannot communicate directly.

In case a node is compromised, key revocation becomes important. This has to be done network-wide and mainly is initiated by controller nodes. Naturally, after the keys associated with the compromised node are deleted from the network, key discovery and path establishment phases are repeated. Other schemes build on the basic random scheme in different ways such as reforming the manner through which the system reacts upon the detection of a compromised node. An example scheme is the multipath key reinforcement scheme. If a node is found to be compromised, multiple independent secure links/paths should be established in order to rekey the network and essentially avoiding any already compromised links while doing so.

3.1.2.2 Q-Composite

This scheme [83] is an extension to the random basic scheme. The extension is mainly concerned with the number of keys needed in order to establish a secure communication link between two nodes. In order to consider two nodes to be connected, q common keys need to overlap between those nodes. This entails that the size of the pool of keys should be reduced. The challenge in this scheme is the ability to find a trade-off between the pool size and the value of q in order to provide an optimal q-composite scheme for a given architecture.

The Q-Composite scheme provides higher resilience to node capture as opposed to the basic scheme is only a small number of nodes are to be captured. Some simulations show that if q is set to 2 in a 10k nodes system, the communication channels compromised if 50 nodes are compromised is 4.74%

against 9.52% for the basic scheme [93]. However, if a larger group of nodes are compromised, q-composite results in a larger portion of the network to be exposed as opposed to the basic scheme.

3.1.2.3 Random Pairwise Key

In order to overcome the scalability issue in the pairwise key scheme, the random pairwise key scheme [83] was introduced. In such a scheme, a given node does not need to store n-1 keys where n is the network size. However, only the equivalent of *(n * p)* keys need to be stored in the key ring of each node where *p* is the probability that two nodes can communicate in a secure manner.

After deployment, nodes start broadcasting their identities in order to determine which nodes share common keys and if so they perform encrypted handshakes. Rebroadcasting to more than one hop can also be carried out. Moreover, a voting mechanism is used in case a given node suspects that another node is compromised. The doubting node casts a vote against the suspect node and if the vote surpasses a given threshold then the suspect node loses its secure communication channels with all the nodes that voted against it iteratively.

3.1.2.4 Polynomial Pool-based

In [89] a scheme that establishes pairwise keys among nodes through the generation of a pool of random bivariate t-degree polynomial *f(x,y)* over the finite field F_q and their distribution is presented. The variable *q* is assumed to be a large primary number that can hold a suitable cryptographic key where also *f(x,y) = f(y,x)*. Each node is naturally given a unique ID that is used by the server to compute a given polynomial share for that node, i.e., *f(ID, y)*. A common key *f(a,b)* for nodes *a* and *b* can be computed when node *a* evaluates *f(a,y)* at point *b* and similarly for node *b*. Randomly generated polynomials are assigned a unique identifier each and every node is loaded with a subset from the pool of polynomials. After deployment, nodes try to establish pairwise keys firstly through direct communication. If certain nodes could not succeed in that then they can use other mediator nodes.

3.1.2.5 Location/Deployment Aware

Several key predistribution schemes that benefit from extra information, e.g., node locations, obtained a priori to actual deployment have been proposed [82], [87], [84]. (Certain schemes separate between the planned/intended position and the real position the nodes end up occupying. This often occurs when nodes are spread using an airplane for instance. Nodes are usually placed on a grid divided into equal size groups each with a sub-key pool from the main key pool. Location information can help in enhancing key connectivity within a group of nodes and between groups.

3.1.3 Dynamic Key Management

What makes this class of key management somehow different is that rekeying here is done periodically or upon detection of a compromised node or on demand in a network that could be heterogeneous. The emphasis here is on an efficient rekeying scheme that can handle more often node addition and key generation.

Exclusion-based systems (EBSs) [85] are proposed for dynamic key management. EBS assigns each node with *k* keys from a pool of keys that has *k + m* keys. Rekeying occurs naturally upon the detection of a node capture or periodically where replacement keys are generated, encrypted with the *m* keys unknown to the compromised nodes and eventually distributed to all nodes aware of the *m* keys. A basic disadvantage in EBS is the possibility that a small number of compromised nodes can conspire and disclose the network key especially if the value of *m* is low. An enhancement on basic EBS was

proposed in [94]; namely, SHELL which clusters the system into a hierarchy and makes use of location information besides basic EBS at each cluster. Another EBS-based scheme; namely, LOCK, presented in [85] does not assume deployment information knowledge and organises the system into three-tiers. A base station lies on the top followed by cluster heads and at the bottom, nodes that include especially dedicated key generation nodes exist. LOCK also employs two layers of administrative EBS keys. Generally, dynamic key management schemes focus on efficient rekeying for networks expected to operate for a long period of time.

3.1.4 Hierarchical Key Management

Hierarchical key management schemes are influenced by the network architecture they try to deal with. A representative scheme called LEAP [95] proposes a key management system designed with the nature of WSNs in focus. LEAP is designed to support in-network processing which is an important feature since WSNs are usually concerned with data aggregation. In-network processing allows nodes to alter, filter and process data before it reaches the final destination which is a promising feature in terms of energy-efficiency. Also, LEAP is motivated by the assumption that different types of messages are exchanged in WSNs and each should require different security measures. Consequently, LEAP provides four types of keys; namely, individual key, pairwise key, cluster key and group key:

- **Individual key:** this key is shared between the nodes and the base station. It is mainly used to compute the message authentication code for the sensed readings. Also, it can be used for alerts sent to the base station. The base station naturally can also use this key to send keying materials or certain commands to specific nodes.
- **Pairwise key:** every node has a pairwise key shared with every immediate neighbour. The pairwise key is used to enable privacy and source authentication such as in cluster key forwarding.
- **Cluster key:** a single key that is shared between a given node and all its neighbours. Its main use is to support in-network processing. For example, in a response to a minimum aggregation command, a node can decrypt a reading forwarded by a neighbour and decide not to add its own if it is not lower than the received neighbour reading.
- **Group key:** this key is shared globally among all nodes and the base station. It is used when there is no advantage in encrypting separately for each node, i.e., in broadcasting.

Individual keys are generated and loaded into the nodes a priori to deployment. This is achieved using a pseudo-random function based on a master key only known to the controller and seeded with node's identity. Moreover, LEAP assumes a lower bound interval of T_{min} that an adversary will need in order to compromise a node besides a time interval $T_{est} < T_{min}$ that is needed by a new node to discover its one-hop neighbours. The interval T_{min} is used in the pairwise key establishment where each node and after a trigger fixed to T_{min} deletes all relevant information that was used during the neighbour discovery and authentication phase. However, each node retains its locally generated master key that it used originally to establish pairwise keys with its neighbours and indeed, the pairwise keys are also kept.

Following pairwise key establishment, cluster key establishment begins. This phase is carried out in a straightforward manner as a given node has already shared pairwise keys with all its neighbours. A given node *n* hence generates a random key (cluster key) and sends it to each neighbour encrypted with the pairwise key it shares with. Each neighbour then transmits its own cluster key back to *n*. Indeed when one neighbour is revoked, *n* needs to reinitiate the previous phase.

Group key establishment on the other hand is a more complex phase as it requires that each node in the network receives the group key originating from the controller. In order for LEAP to achieve that, it uses

μ-TESLA [90] for message broadcast authentication and assumes the existence of a routing protocol that organises the network as a spanning tree. Moreover, it is essential that each message is authenticated before it is forwarded or processed, hence, a local broadcast authentication is provided by LEAP. This local broadcast authentication does not require time synchronization among neighbouring nodes but uses a one-way key chain approach [88].

3.1.5 Suitability Discussion

In the light of pSHIELD's general requirements to potentially handle security in lightweight devices, a key management system needs to be suitable accordingly. Issues such as limited storage, computational and energy constraints need to be kept in mind while selecting or designing a key management scheme that deals with lightweight devices. The architecture of the system to which security has to be provided is also an essential input to the key management selection process. The application prototype in pSHIELD has been identified to be a relatively small scale hierarchical system in a train carriage. The system consists of lightweight sensor devices and a local controller (cluster head) that could communicate with some main server.

Conventional key management schemes, while seem to be easy to deploy, each suffer from drawbacks that could make an adversary attack easier. In a single network-wide scheme, less computational, communication and storage resources are needed, however, it suffice only for one attacker to compromise one node for the whole network to be compromised. A more distributed scheme such as the pairwise key scheme provides more resilience to attacks that could happen in a single network-wide key scheme. However, pairwise key scheme faces an overhead as each node needs to establish a unique key with every other node. This incurs storage and communication problems as the network grows in size, however, for small networks, a pairwise key scheme as the base for a more complete key management scheme could be beneficial. Moreover, in a completely centralised key management scheme, node capture attacks and memory requirements are minimized but the scheme faces scalability and single point of failure issues.

More sophisticated key management schemes have evolved recently. Several key pre-distribution schemes deploy probability approaches in order to preload nodes with cryptographic keys. In the basic scheme, given a key pool size of 10k keys, it would require that each node stores 75 keys in order to have a probability $p = 0.5$ that two nodes share a key. This scheme could provide a scalable key management scheme and would certainly be more suitable for bigger networks, however, for small scale networks with known deployment locations it would not make full sense. Also, the basic scheme lacks node-to-node authentication. In addition, memory constrained nodes have usually 4KB of memory while 75 keys amount to (75 x 160 bits = ~1.46KB) for an ECC-160 cryptographic algorithm per example which is a considerable memory requirement. Moreover, q-composite, an enhancement on the basic scheme, introduces more conditions on the path establishment process which ameliorate communication links and node capture resilience. However, it still cannot cope with large-scale attacks or node capture.

A scheme based on the conventional pairwise scheme adds an element of randomness to the original process, i.e., the random pairwise scheme. Compared to the basic and q-composite schemes, this scheme adds more resilience to node capture and node replication attacks given the unique key each node holds. However, the scheme has a scalability issue. On the other hand, the polynomial pool-based scheme supports an undetermined growth in the network size after deployment while continuously being resistant to t collisions, i.e., if t or more polynomials are broken, then the network can be compromised.

Dynamic key management is being explored for better flexibility in terms of networks growing over time and for longer network life as opposed to key pre-distribution. Also, they tend to use smaller key pools and more efficient rekeying where also each node holds fewer keys. It is an interesting direction for key management, however challenging, and it is under growing research.

The hierarchical key management scheme is the natural response to systems following such an organisation, i.e., hierarchical in terms of the existence of a powerful server, cluster heads/controllers and nodes. Such a scheme coupled with the support of security in the presence of lightweight devices could be a good option for the proposed application in the pSHIELD project. LEAP provides an interesting approach to dealing with several types of attacks by distinguishing among the different security needs of different network traffic. This is carried out through the provision of different types of keys on different levels and through the leveraging of μ-TESLA (see Section 3.2.1). In addition, LEAP has the ability to adopt different cyphering techniques and can be considered a promising approach for hierarchical designs such as in the application presented in the pSHIELD project.

3.1.6 The Controlled Randomness Protocol

An embedded system can incur an interesting trade-off on security level and resource consumption. From a security point of view, the keys must be often refreshed, as explained earlier, in order to maintain the required security level. From a system resource consumption point of view, the keys must be rarely changed, in order to minimize the consumption of precious resources (processor, power and bandwidth). Further, in some usage scenarios, advanced care must be taken in order to ensure that the new keys will be available by the time they must be used, especially when only intermittent connectivity exists.

The “controlled randomness protocol” (CRP) for cryptographic key management is proposed as an improvement for the security level of secure communication protocols. The CRP allows multiple keys to be valid at any given time; it neither alters the total number of keys needed in the underlying cryptographic algorithms, nor the need of a control channel to periodically refresh keys. However, the increased security offered by CRP allows for far less frequent key exchanges. Details on the design and implementation of the protocol can be found on D3.4 “SPD self-x and cryptographic technologies”

3.2 Authentication

Authentication, in abstract, is the ability to verify a given entity's identity. In terms of security, cryptography is one of the main means to authenticate a sender's identity or the originality of given message. Public key cryptography, discussed in [92], particularly is the typical approach to authenticate nodes usually through digital signatures or certificate authority. However, as such a cryptographic paradigm can be expensive in terms of resources in lightweight devices; different symmetric-cryptography-based approaches are being used (see Section 8 in [92]). Here, a representative lightweight broadcast authentication protocol is presented.

3.2.1 μ-TESLA

Based on the TESLA [90] protocol for broadcast authentication, μ-TESLA is an enhanced version that alters TESLA to deal with lightweight devices. It essentially emulates asymmetry through the disclosure of delayed symmetric keys. This is a more efficient operation manner than that usually required by public key cryptography. It also reduces communication overhead by only sending keys once every epoch as opposed to sending them with every packet as in TESLA. Also, μ-TESLA restricts the number of authenticated senders in order to reduce key chain storage needs at nodes.

In μ-TESLA, the base station needs to be loosely synchronised with the nodes with an error margin known a priori. In order to send a packet, the base station first has to compute the message authentication code (MAC) using the cryptographic hash function of choice and a key that is secret at that point of time. Once a given node receives the packet, it can determine that the MAC keys used for that packet has not been disclosed yet based on the synchronization mechanism in place and on the estimated time the keys are disclosed. The node stores the packet in its buffer until the key is disclosed by the base station after which it can verify the correctness of the key and decrypt the buffered packet. A node can verify the correctness of the key from the chain of keys using a one-way function *F*. Nodes are expected to be aware of the commitment that is the first key in the chain, i.e., K_0 where $K_i = F(K_{i+1})$. Each key in the chain is issued for a specific time interval and all packets sent during that interval are encrypted using that same key. Evidently, the key is sent in a special packet periodically and in an independent manner from usual packets sending. For example, assume that a sender discloses the key it used to encrypt a current packet after 2 time intervals from sending it. In order to decrypt packet P_1 sent in time interval *1*, the receiver should wait until the third interval in order to get K_1. Even if K_1 was lost for some reason, the receiver can benefit from K_2 sent in the fourth interval to decrypt P_1 by calculating $K_1 = F(K_2)$ and verifying its correctness by comparing its commitment (K_0) to the deduced $F(F(K_2))$. Once K_1 is verified it can then be used to decrypt P_1.

Indeed, μ-TESLA needs to incorporate a cyphering algorithm that suits the needs of lightweight nodes. Promising results were shown when block cyphers are used in the counter mode. It is also claimed that the energy spent on security is marginal as opposed to that used in sending and receiving messages. Also, as data authentication, freshness and confidentiality requirements amount to an overhead of 6 bytes out of some 30 bytes packet, it is argued that it could be feasible to satisfy those requirements per packet.

4 Intrusion Detection on Wireless Sensor Networks

In the world of communication a trusted connection is an essential requirement for the final user and system administrators of a system. Due to this designing prerequisite a trusted connectivity in a system is vital to obtain a high quality service. This section is devoted to the study of the requirements for lightweight link-layer secure communication in wireless sensor network scenarios. For that purpose specific intrusion detection systems are studied. How data reliability mechanisms (including confidentiality, integrity, and authenticity) can be shared between different wireless network technologies are also investigated.

4.1 Wireless Sensor Networks

A wireless sensor networks (WSN) consists of spatially distributed autonomous sensor to cooperatively monitor physical or environmental conditions. In addition to one or more sensors, each node in a sensor network is a small and simple computer typically equipped with a radio transceiver or other wireless communication devices, a small micro-controller, and an energy source, usually a battery. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth.

A more detailed WSN description is given in [14]. Sensors integrated into structures, machinery, and the environment, coupled with the efficient delivery of sensed information, could provide tremendous benefits to society. Potential benefits include: fewer catastrophic failures, conservation of natural resources, improved manufacturing productivity, improved emergency response, and enhanced homeland security [15]. However, barriers to the widespread use of sensors in structures and machines remain. Bundles of lead wires and fiber optic “tails” are subject to breakage and connector failures. Long wire bundles represent a significant installation and long term maintenance cost, limiting the number of sensors that may be deployed, and therefore reducing the overall quality of the data reported. Wireless sensing networks can eliminate these costs, easing installation and eliminating connectors.

The ideal wireless sensor is networked and scalable, consumes very little power, is smart and software programmable, capable of fast data acquisition, reliable and accurate over the long term, costs little to purchase and install, and requires no real maintenance.

Selecting the optimum sensors and wireless communications link requires knowledge of the application and problem definition. Battery life, sensor update rates, and size are all major design considerations. Examples of low data rate sensors include temperature, humidity, and peak strain captured passively. Examples of high data rate sensors include strain, acceleration, and vibration.

Recent advances have resulted in the ability to integrate sensors, radio communications, and digital electronics into a single integrated circuit (IC) package. This capability is enabling networks of very low cost sensors that are able to communicate with each other using low power wireless data routing protocols. A wireless sensor network generally consists of a base station (or “gateway”) that can communicate with a number of wireless sensors via a radio link. Data is collected at the wireless sensor node, compressed, and transmitted to the gateway directly or, if required, uses other wireless sensor nodes to forward data to the gateway. The transmitted data is then presented to the system by the gateway connection. The purpose of this chapter is to provide a brief technical introduction to wireless sensor networks and present a few applications in which wireless sensor networks are enabling.

There are three wireless networks categories: cellular, ad-hoc and sensor. These networks have a lot of common characteristics, but they also have some significant differences. In ad-hoc networks, each node share information with the rest of the network, in contrast with cellular network where the task of sharing data is only done by some specific nodes. There are some key differences between ad-hoc networks and Wireless Sensor Networks: the availability of energy and computational resources are limited in WSN and there is a higher risk that nodes are compromised due to the fact that WSN are not hold by humans [16].

A sensor network normally constitutes a wireless ad-hoc network, meaning that each sensor supports a multi-hop routing algorithm where nodes function as forwarders, relaying data packets to a base station.

Wireless sensor network usual characteristics include:

- Limited power they can harvest or store.
- Ability to withstand harsh environmental conditions.
- Ability to cope with node failures.
- Mobility of nodes.
- Dynamic network topology.
- Communication failure.
- Heterogeneity of nodes.
- Large scale of deployment.
- Unattended operation.
- Node capacity is scalable, only limited by bandwidth of gateway node.

As it is explained in [16], each sensor node in a WSN is equipped with a sensor, processing unit (in most cases a microcontroller), memory unit, wireless transceiver and battery. There are different types of sensors such as thermal, mechanical, chemical, optical, acoustic, and they monitor variety of physical parameters such as temperature, radiation, barometric pressure, ambient light, movement, sounds, humidity, etc. Sensed data are sent to gateway nodes (also called sinks or base stations) that interface the WSN to the external world, in most cases the Internet.

[15] describes, also, a WSN architectural classification. At this point, there are three different types of topologies described:

- **Star Network (Single Point to Multi-point).** A star network is a communications topology where a single base station can send and/or receive a message to a number of remote nodes. The remote nodes can only send or receive a message from the single base station, they are not permitted to send messages to each other. The advantage of this type of network for wireless sensor networks is in its simplicity and the ability to keep the remote node's power consumption to a minimum. It also allows for low latency communications between the remote node and the base station. The disadvantage of such a network is that the base station must be within radio transmission range of all the individual nodes and is not as robust as other networks due to its dependency on a single node to manage the network.

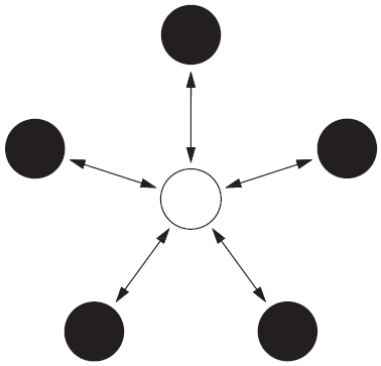

Figure 4-1: Star Network Topology

- **Mesh Network.** A mesh network allows for any node in the network to transmit to any other node in the network that is within its radio transmission range. This allows for what is known as multi-hop communications; that is, if a node wants to send a message to another node that is out of radio communications range, it can use an intermediate node to forward the message to the desired node. This network topology has the advantage of redundancy and scalability. If an individual node fails, a remote node still can communicate to any other node in its range, which in turn, can forward the message to the desired location. In addition, the range of the network is not necessarily limited by the range in between single nodes, it can simply be extended by adding more nodes to the system. The disadvantage of this type of network is in power consumption for the nodes that implement the multi-hop communications are generally higher than for the nodes that don't have this capability, often limiting the battery life. Additionally, as the number of communication hops to a destination increases, the time to deliver the message also increases, especially if low power operation of the nodes is a requirement.

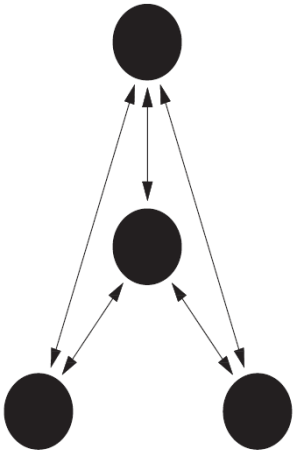

Figure 4-2: Mesh Network Topology

- **Hybrid Star – Mesh Network.** A hybrid between the star and mesh network provides for a robust and versatile communications network, while maintaining the ability to keep the wireless sensor nodes power consumption to a minimum. In this network topology, the lowest power sensor nodes are not enabled with the ability to forward messages. This allows for minimal power consumption to be maintained. However, other nodes on the network are enabled with multi-hop capability, allowing them to forward messages from the low power nodes to other nodes on the network. Generally, the nodes with the multi-hop capability are higher power, and if possible,

are often plugged into the electrical mains line. This is the topology implemented by the up and coming mesh networking standard known as ZigBee.

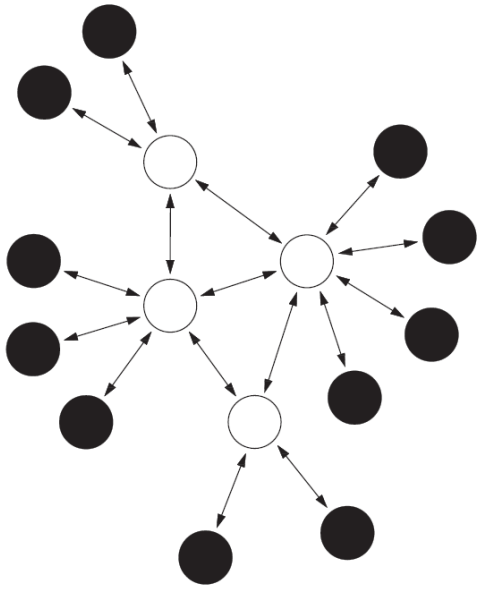

Figure 4-3: Hybrid Star-Mesh Topology

The physical radio layer defines the operating frequency, modulation scheme, and hardware interface of the radio to the system. There are many low power proprietary radio integrated circuits that are appropriate choices for the radio layer in wireless sensor networks. If possible, it is advantageous to use a radio interface that is standards based. This allows for interoperability among multiple companies networks. A discussion of existing radio standards and how they may or may not apply to wireless sensor networks is given below. [15] analyses the different standards use in communications in WSN:

- **IEEE802.11x.** IEEE802.11 is a standard that is meant for local area networking for relatively high bandwidth data transfer between computers or other devices. The data transfer rate ranges from as low as 1 Mbps to over 50 Mbps. Typical transmission range is 300 feet with a standard antenna; the range can be greatly improved with use of a directional high gain antenna. Both frequency hopping and direct sequence spread spectrum modulation schemes are available. While the data rates are certainly high enough for wireless sensor applications, the power requirements generally preclude its use in wireless sensor applications.
- **Bluetooth (IEEE802.15.1 and .2).** Bluetooth is a personal area network (PAN) standard that is lower power than 802.11. It was originally specified to serve applications such as data transfer from personal computers to peripheral devices such as cell phones or personal digital assistants. Bluetooth uses a star network topology that supports up to seven remote nodes communicating with a single base station. While some companies have built wireless sensors based on Bluetooth, they have not been met with wide acceptance due to limitations of the Bluetooth protocol including: 1) Relatively high power for a short transmission range. 2) Nodes take a long time to synchronize to network when returning from sleep mode, which increases average system power. 3) Low number of nodes per network (<=7 nodes per piconet). 4) Medium access controller (MAC) layer is overly complex when compared to that required for wireless sensor applications.
- **IEEE 802.15.4.** The 802.15.4 standard was specifically designed for the requirements of wireless sensing applications. The standard is very flexible, as it specifies multiple data rates and multiple transmission frequencies. The power requirements are moderately low; however, the

hardware is designed to allow for the radio to be put to sleep, which reduces the power to a minimal amount. Additionally, when the node wakes up from sleep mode, rapid synchronization to the network can be achieved. This capability allows for very low average power supply current when the radio can be periodically turned off. The standard supports the following characteristics: 1) Transmission frequencies, 868 MHz/902–928 MHz/2.48–2.5 GHz. 2) Data rates of 20 Kbps (868 MHz Band) 40 Kbps (902 MHz band) and 250 Kbps (2.4 GHz band). 3) Supports star and peer-to-peer (mesh) network connections. 4) Standard specifies optional use of AES-128 security for encryption of transmitted data. 5) Link quality indication, which is useful for multi-hop mesh networking algorithms. 6) Uses direct sequence spread spectrum (DSSS) for robust data communications.

It is expected that of the three aforementioned standards, the IEEE 802.15.4 will become most widely accepted for wireless sensing applications. The 2.4-GHz band will be widely used, as it is essentially a worldwide license-free band. The high data rates accommodated by the 2.4-GHz specification will allow for lower system power due to the lower amount of radio transmission time to transfer data as compared to the lower frequency bands.

- **ZigBee.** The ZigBee™ Alliance is an association of companies working together to enable reliable, cost-effective, low-power, wirelessly networked monitoring and control products based on an open global standard. The ZigBee alliance specifies the IEEE 802.15.4 as the physical and MAC layer and is seeking to standardize higher level applications such as lighting control and HVAC monitoring. It also serves as the compliance arm to IEEE802.15.4 much as the Wi-Fi alliance served the IEEE802.11 specification. The ZigBee network specification, to be ratified in 2004, will support both star network and hybrid star mesh networks. The ZigBee alliance encompasses the IEEE802.15.4 specification and expands on the network specification and the application interface.

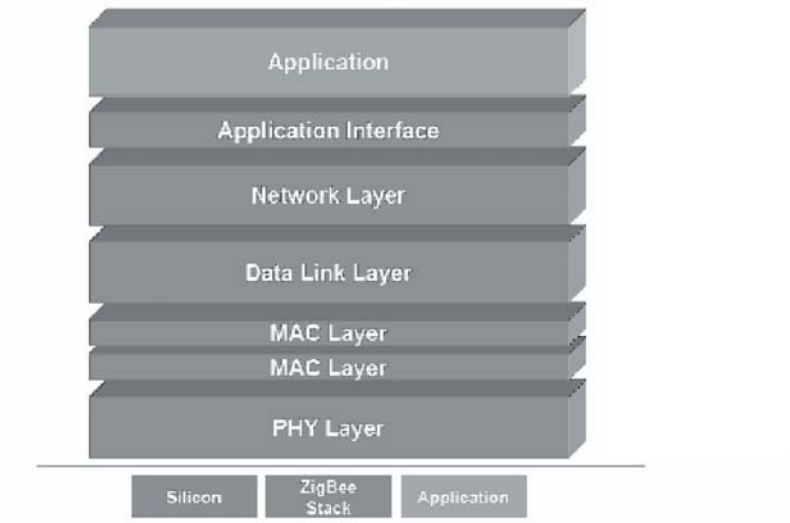

Figure 4-4: Zigbee Structure

- **IEEE1451.5** While the IEEE802.15.4 standard specifies a communication architecture that is appropriate for wireless sensor networks, it stops short of defining specifics about the sensor interface. The IEEE1451.5 wireless sensor working group aims to build on the efforts of previous IEEE1451 smart sensor working groups to standardize the interface of sensors to a wireless network. Currently, the IEEE802.15.4 physical layer has been chosen as the wireless networking communications interface.

Several standards are currently either ratified or under development for wireless sensor networks. There are a number of standardization bodies in the field of wireless sensor networks. The IEEE focuses on the physical and MAC layers with the IEEE 1415 standard; the Internet Engineering Task Force works on layer 3 and above. In addition to these, bodies such as the International Society of Automation

provide vertical solutions with the ISA 100 standard, covering all protocol layers. Finally, there are also several non-standard, proprietary mechanisms and specifications.

When designing a wireless sensor networks important areas like power management, synchronization, localization and wake-up must be taken in account. Involving all this areas should guarantee the secure and correct performance of the network itself.

4.1.1 WSN Threats and Solutions

This section tries to identify the security risks in Wireless Sensor Networks. Once that the risk has been set as a weak point, a solution or security system will be proposed to fix the security issue. Those risks would be identified using related security works.

4.1.1.1 WSN General Security Attacks

There are four aspects of a wireless sensor network that security must protect:

A. Confidentiality

B. Data Integrity

C. Service Availability

D. Energy

Confidentiality, Data Integrity and Service Availability are addressed by security systems in wired networks, but Energy is unique to the wireless sensor networks due to the resource limitation constraint. A short explanation of these categories of attacks will be given in the following lines.

A. Stealing Data (Confidentiality): In electronic systems, it is necessary to protect the content inside all sent messages from being figured out by enemies or actors that do not belong to the system. Because of the wireless nature of the WSN, it is easy for those actors to listen in on all the messages sent in the network, so to maintain confidentiality, the network must encrypt all the messages. One of the most popular encrypting solutions today is public-key encryption. This is very powerful because it allows one to receive encrypted messages without even sharing a secret key with the sender. It must be taken into account the characteristics of WSN in terms of processing capacity and energy source limitation, so the selected encryption algorithm must me lightweight enough to fulfill all the requirements.

B. Altering/Generating False Data (Data Integrity): Due to sensor networks are used to monitor environments, data integrity is even more important than confidentiality. If attackers are able to make the data collected by the WSN incomplete or incorrect, the administrator of the WSN will not probably know what is really going on in the monitored environment. In other networks, the same asymmetric key system that is used for encryption can be used for digital signatures, but this requires a lot of additional overhead. The signature may consist of a lot of additional bytes of data added on to a transmission (which takes additional energy), and verifying the signature can be very computationally expensive. Clearly, different techniques are needed in WSNs.

C. Attacks on Service Availability: The goal in these kinds of attacks is to make the network not function properly. This can be done by sending bogus routing information (for example advertising a route that does not exist). It can also be done by flooding the network with packets (denial of service attack), or even jamming the frequency at the physical layer. Another interesting type of attack is homing, where the attacker looks at network traffic to deduce the geographic location of critical nodes, such as cluster heads or neighbors of the base station. The attacker can then physically disable these nodes. This leads to another type of attack: the "black hole attack". In a "black hole" attack, the attacker compromises all the neighbors of the base station, making it effectively a black hole. A final kind of attack on service availability is a de-synchronization attack, where the attacker tries to disrupt a transport-layer connection, by forging packets from either side.

D. Denial of Service Attacks (Energy): Due to general WSN characteristics, the energy constraint adds a new element that can complicate security issues. The energy amount limitation results in a necessity to make sure that the system does not waste energy listening to or re-transmitting unnecessary or bad packets. This introduces a whole new set of possible attacks. These include constantly sending RTS packets to stop nodes from going to a low power "sleep" state, sending falsified or repeated packets so that nodes waste energy re-transmitting them, or draining the power of a node by forcing it to do excessive computations.

4.1.1.2 Routing Threats in Wireless Sensor Networks

There are some works where many types of attack on routing protocols in detail are studied and affection on common routing protocols in WSNs. Those works assume that there are two types of attacks, outside attacks and inside attacks. It is considered outsides attack can be prevented through the use of link layer security mechanisms like encryption. In [17], two types of attackers are proposed, mote-class attacker and laptop-class attacker. In mote-class, the attacker has access to a few sensor nodes with similar capabilities as the legitimate nodes. These nodes are tampered and reprogrammed for attacker's purpose. In laptop-class, the attacker has access to more powerful devices like laptop with greater battery power, high CPU processing and high-power radio transmitter. In this case, the attacker has an advantage to deploy attacks on the network. Most common network layer attacks on WSNs are explained in the following lines pointing out the characteristics of these attacks. As it is mentioned in [17], many sensor network routing protocols are quite simple, and for this reason are sometimes even more susceptible to attacks against general ad-hoc routing protocols. Most network layer attacks against sensor networks fall into one of the following categories:

A. Selective forwarding: In selective forwarding attack, malicious nodes try to stop routing information in the sensor networks by refusing to forward or drop the messages pass through them. Another trend of this attack is that the malicious nodes may forward the messages to the wrong path, creating unfaithful routing information in the network.

B. Sinkhole attacks: In sinkhole attack, the attacker lures nearly all the traffic from the particular area through a malicious node, creating a metaphorical sinkhole. The laptop-class attacker may use higher computation and communication power than a legitimate node to advertise itself as a shortest path to base-station or cluster head in our case.

C. Wormhole attacks: In wormhole attack, the attacker will tunnel messages received in one malicious node and replay them in a different part of the network. Wormhole attacks more commonly involve two distant malicious nodes colluding to understate their distance from each other by relaying packets along an out-of-bound channel available only to attacker. The two malicious nodes usually pronounce that their distance is just two hops away from the base station.

D. Hello flood attacks: Many routing protocols use Hello broadcast messages to announce themselves to their neighbor nodes. The nodes received Hello messages assume that source nodes is within the radio range and add source node in their neighbor list. The laptop-class attacker can broadcast Hello messages with large transmission power to convince a group of nodes that they are neighbors.

E. Sybil attacks: In this attack, a malicious node can present multiple identities to other nodes in the network. Sybil attack poses a significant threat to most of geographic routing protocols. The Sybil attack can significantly reduce the effectiveness of fault-tolerant schemes such as distributed storage, spread and multipath routing, and topology maintenance. Sybil attacks can be prevented through the use of link layer authentication [98].

F. Spoofing: In this attack, an intruder wants to establish itself as a legitimate node. To achieve its goal, the malicious or foreign node will probably try to copy the ID or MAC address of a legitimate node of the network. After entering the network, the malicious node will have the chance to behave like a normal node. Spoofed, altered, or replayed routing information target the routing information exchanged between nodes, and by this method adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages, partition the network, increase end-to-end latency, etc.

G. Acknowledgement spoofing: Several sensor network routing algorithms rely on implicit or explicit link layer acknowledgements. Due to the inherent broadcast medium, an adversary can spoof link layer acknowledgements for “overhead” packets addressed to neighbouring nodes. Goals include convincing the sender that a weak link is strong or that a dead or disabled node is alive.

Major classes of attacks not countered by link layer encryption and authentication mechanisms are wormhole attacks and HELLO flood attacks.

There are other several groups looking into the architecture of wireless sensor embedded networks. For example, [18] has focused in on architectural support for system-level optimization, enabling application specific optimization of communication protocols. By providing tight coupling between application and protocol level processing it allows application-specific implementations of traditional protocols. Those protocols can expose as much or as less information up into the application as they want.

[19], instead, is a project that gives a different WSN architecture. It describes a generic security package that developers can integrate into sensor network applications, addressing security in devices where energy and computing present significant resource limitations. TinySec, based in existing security primitives that other researchers have proven to be secure, is a lightweight and efficient link layer security protocol that is tailored to sensor networks. TinySec shows that, with sufficient engineering

effort, it is possible to encrypt all communications entirely in software without major performance degradation, with no need of hardware.

4.2 Energy Assessment

This section describes the experiments performed in order to analyse the energy footprint of the mobile ad hoc networks. Energy consumption is a major security concern in embedded systems but consumption information is often missing. For the experiment, the selected protocol was Random-Walk Gossip (RWG).

We know that display, radio transceivers and CPU are the main causes of battery discharge. We performed the experiments on Nokia N97since it can be used with an accurate measurement tool (NEP) [105]. We used the RWG Client on top of the standard software on the device and then tried to isolate the impact of the protocol operation as follows.

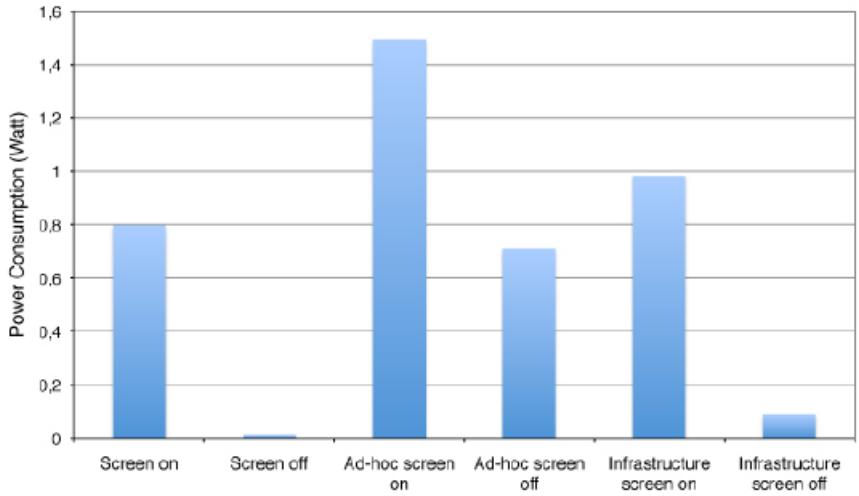

Figure 4-5: Power consumption of Nokia N97

First, we measured the power consumption on the Nokia N97 in different states (the protocol was not running). Figure 4-5 shows in the two leftmost bars that switching on the screen consumes 0,79 W. The content of the screen was the application menu, which does not have any graphical activity. The implementation of the protocol uses the WLAN interface in ad hoc mode, which means that the power consumption will be around 0,7 W when the protocol is running. This is shown by the 4th bar from the left in Figure 4-5. Note that the ad hoc mode consumes more energy than infrastructure since the node is listening to the channel all the time and uses less power saving mechanisms. This is shown in the rightmost bar.

Second, the energy consumption when running the protocol was studied in the following experiments. The RWG Client was running on top of the protocol and as stated before it was using the WLAN interface in ad hoc mode. The transmission power was 100mW by default.

In idle state, without sending any data, the most noticeable increase in power consumption was due to using the WLAN interface. As stated in Subsection IV-A, the CPU usage in idle state is 0% and it increases to around 1% when the mechanism that sends a packet every once in a while performs its

duty. The following tests verify the impact of that mechanism in idle state. First, RWG was running in idle state without sending any message and the battery (1500 mAh) was discharged after 7:27 hours. In the second, the protocol sent a message every second and the battery lasted 7:18 hours. Therefore, we can conclude that the impact of the mechanism on the lifetime of the idle state is only 2%, not affecting significantly the energy consumption. Consequently, the impact of the protocol on the consumed energy in idle state is due to the use of the WLAN interface.

In operation state, two Nokia N97 were used and a message was sent from one device every second. Our intuition was that the use of more memory can lead to more CPU load, which consumes more energy. Therefore, the test was performed with the message buffer of the phones empty as well as with 500 messages to show the impact of message storage on energy consumption. The energy consumption difference was not significant. Thus, we conclude that the implementation of RWG handles the messages an energy-efficient manner.

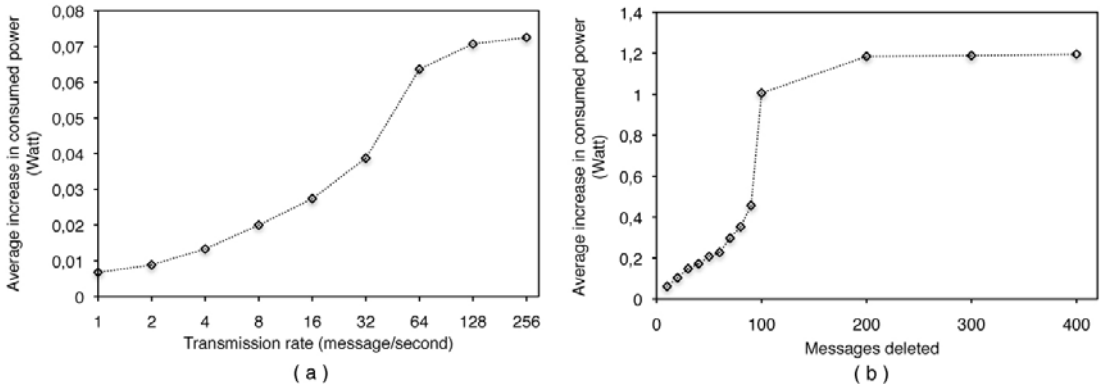

Figure 4-6: Average increase in consumed power due to (a) data rate in logarithmic scale and (b) a CPU demanding operation. The reference value is the consumed power in protocol's idle state.

Third, the power consumption increase due to data rate was tested. The consumed power in the idle state (WLAN active in ad hoc mode) was taken as reference value. The RWG Client was used to send messages at different transmission rates. The size of the packets was 98 bytes, including MAC, IP, UDP and RWG headers. The average increase in consumed power of the sending period is shown in Figure 4-6(a), which shows that, as expected, the consumed power increases when the message transmission rate increases. However, the average increase in consumed power is very small in comparison with the 0,7 W for having the WLAN active in ad hoc mode.

Finally, the average increase in consumed power of some CPU demanding operations was tested. One of the most consuming operations is deleting many messages from the buffer at the same time. The test consisted of deleting different number of messages from the buffer at the same time when their TTL expired. Figure 4-6(b) shows that the increase in consumed power converges to a maximum when deleting more messages. This maximum is reached when the CPU load is 100%. Deleting 400 messages consumes more energy than 300 although the consumed power level is the same but it lasts longer. Note that with the radio being on by default due to the ad hoc mode the quantitative increase is larger due to higher CPU usage (Figure 4-6(b)) than due to higher transmission rate (Figure 4-6(a)).

To summarize, one could conclude that the energy consumption footprint of the implementation of the RWG protocol is mostly due to the use of the WLAN interface in ad hoc mode, which is more significant than the other aspects.

4.3 Intrusion Detection Systems

Nowadays, communication devices have become an essential part of most people in the world, affecting every single part of daily life like studies, health, business world, entertainment, etc. This situation has given birth to a new brand of application using connections as an important feature. Furthermore, and as a logical part of communications, a lot of information is shared in these networks. This scenario may contain security holes, due to economical reasons or neglect, which could open the door to intrusions. An intrusion is defined as a violation of the security policy of a system. The main target of these attacks or intrusions is usually the extraction or modification of the information sent to derivate in unexpected behaviours in the networks.

There are several security methodologies that are focused in the prevention of common security attacks using special hardware or software. However, these methods usually cover a single or a few types of attacks, so that there are many times when second or extra security lines are needed, at least to identify and warn the intrusions or attacks in the system. An Intrusion Detection System (IDS) is able to complete the requirements to this second security line that any network needs.

IBM labs in Zurich defines some interesting IDS features like audit source location, methodology of detection, computing location, usage frequency and response to intrusions in [20]. These descriptions may be very useful when understanding IDS properties. As another area of IDS concept, just like explained in [21], the classification or taxonomy of intrusion detection systems has been considered in numerous works from which the ones of Debar et al. [22] and Axelsson [23] could be highlighted. The most common classification is carried out based on three functional characteristics of IDSs:

- **Information Sources:** Referred to the data source used to determine if an intrusion has happened.
- **Analysis:** This is the detection method used. The information gathered in the previous step can be analysed by means of different techniques.
- **Response:** Once an intrusion has been detected, the IDS can answer to that with an active response or just record the event taking any further measures.

The following figure, describes the typical classification of intrusion detection systems.

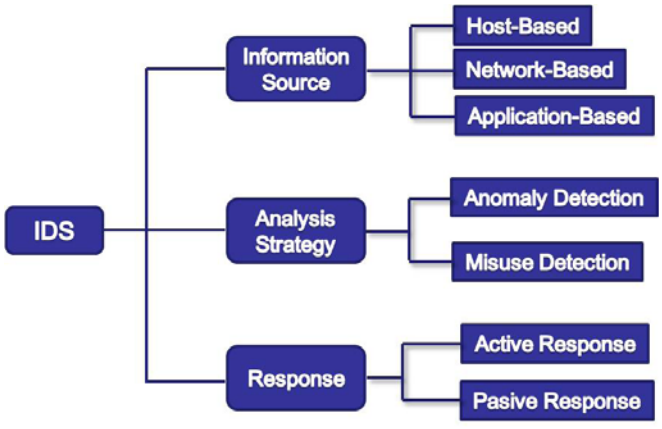

Figure 4-7: IDS classification

4.3.1 Information Sources

From the beginning IDS researches have been working with data coming from diverse sources trying to identify the existence of an intrusion. These data can be divided into three main groups: those obtained from a machine or host (logs or audit trails, system calls, keyboard commands, etc.), those obtained from monitoring a network (TCP, UDP, RTP, ICMP, etc.), and finally, data obtained from the execution of applications (HTTP, DNS, Telnet, FTP, SSH, SMTP, etc.).

4.3.2 Analysis Strategy

The method of detection that IDSs use is referred as analysis strategy. Figure 2.4 in [21] shows a classification of IDSs taking into account the analysis strategy, based on the research made by Noel et al. [24] and Lazarevic et al. [25]. Some of the IDSs derive from other ones. This is represented by doted arrows in the figure. On the other hand, other IDSs are the result of applying different techniques and strategies of analysis, represented by lines.

4.3.2.1 Misuse Detection

A misuse detection based IDS monitors the activities of a system and compares them with signatures of attacks that are stored in a database. When monitored activities coincide with a signature, an alert is produced. Misuse detection makes use of the a priori knowledge of the activities and sequences that compose an attack. With this method, attempts that try to exploit known vulnerabilities or typical attack patterns can be discovered. This is the most extended strategy in commercial IDSs.

Typically, a misuse detection system has two main components [26]: A language or model to describe or represent the techniques used by attackers, and monitoring programs to detect the presence of an attack based on the descriptions or representations given.

The advantage of a misuse detection IDS is the reliability of the detection of known attack patterns. In a similar way to an anti-virus software, the malicious behaviour can be identified with an acceptable accuracy.

On the other hand, the main disadvantage is that the attack pattern must be known in advance, otherwise it will not be detected. Thus, new or unknown attacks will go through the IDS undetected and also, the system can be easily fooled by introducing small variations in known attack patterns. Another disadvantage of this kind of IDSs is that the system must be configured manually to fit the system's characteristics if a low false positive rate is desired. There are several methods to implement this kind of systems:

1. **Knowledge Based Systems**

   These types of methods check the events in hosts or networks looking for predefined attack patterns. The objective is to use representations of known attacks to manage the occurrence of those attacks. There are several ways to represent attacks: expert systems, attack signatures, state transitions, and also the more particular case of Petri nets.

   Expert Systems. These systems codify the knowledge of databases by means of “if-then-else” like implication rules (condition-action). When all conditions are fulfilled, the rule is activated and the consequence of the rule is executed. The inference engine is the one to decide if an intrusion has occurred making use of rules and events. One of the limitations of these systems is that rules are not temporally sequential what makes difficult to specify the steps of intrusions

based on time. Some misuse detectors that use expert systems are NIDX [27], ComputerWatch [28], ISOA [29], AutoGuard (which uses case-based reasoning) [30] and Shadow [31]. Production-Based Expert System Toolset (P-BEST), [32] is a programmable shell for an expert system created by AlanWhitehurst and Fred Gilham. This tool set was used in projects like Multics Intrusion and Alerting System, MIDAS [13], IDES [33], NIDES [34] and also in the signature analysis engine eXpert used by EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances) [35].

Signature Detection. Also known as Model Based Reasoning System, signature detection observes the occurrence of chains (or chain patterns) that could be considered as suspicious. Signature Detection compares the events with the stored chains or signatures of a database of attack scenarios looking for coincidences. These scenarios are stored as sequences of activities or behaviours. Its main disadvantage is the necessity of including new signatures for every new attack or discovered vulnerability. The most popular signature detection based IDSs are Snort [36], Network Flight Recorder (NFR) [37], Network Security Monitor (NSM) [38], NetRanger {[39] (now Cisco Intrusion Detection [40]), NID [41], RealSecure [42] (old version of ISS Proventia, recently acquired by IBM), Computer Misuse Detection System (CMDS ^{TM}) [43], NetProwler is now Intruder Alert (by AXENT, joined to Symantec) [44] and Haystack [45].

State Transition Analysis. Created from the construction of a finite state machine. Attack scenarios are represented as a sequence of transitions that characterize the evolution of the security state of a system. When the machine reaches a state considered as an intrusion, an alert is generated. This technique was initially suggested in STAT (State Transition Analysis Tool) [46], and later other applications such as USTAT (UNIX State Transition Analysis Tool) [47] and NetSTAT (Network Based State Transition Analysis Tool) [48] were developed. All of them were created in the University of California at Santa Barbara.

Coloured Petri Nets are a special case of state transition. They are used in the project IDIOT [26] of Purdue University. The conceptual simplicity, generality and graphical representation are its main advantages. Nevertheless, looking for equivalences between a complex network and audit trails can be computationally expensive. A special type of Petri net is applied in [49], more precisely the Fuzzy Reasoning Petri Nets (FRPN) that are used to represent a base of fuzzy rules and derive the decision of the detection using an inference engine. This is a technique based on the combination of fuzzy logic and the methodology of expert systems.

In [21], there is more information and examples with references of misuse detectors based in knowledge based systems.

2. Machine Learning Based Systems

The machine learning methods used for misuse detection automatically discover and generate attack patterns. There is no need to manually develop predefined patterns or signatures. These methods exploit the regularities or associations inherent to data. The objective is the same, to create representations of attacks, but the difference is that now they are induced automatically avoiding the expensive design of representations as seen previously. To do so, the system starts with a learning phase where known attack sequences are introduced to obtain intrusion patterns. For misuse detection, tagged predictive models are constructed. They can be tagged according to records of attacks tagged as “probe”, “DoS”, “R2L” and “U2R” or simply as “normal” or “intrusive”.

Two very important works in this area are JAM (Java Agents for Metalearning) [50] and MADAM ID (Mining Audit Data for Automated Models for intrusion Detection) [51] projects. Both of them developed by Wenkee Lee and Salvatore Stolfo from Columbia University. JAM uses data mining techniques to discover intrusion patterns and uses several classifiers in the learning phase (meta-learning) to build the misuse detection model. MADAM ID also uses data mining models to develop rules for misuse detection.

A more detailed overview of works that applied data mining techniques to intrusion detection can be found in [52]. These surveys present an exhaustive analysis of systems based on machine learning, for both misuse detection and anomaly detection. A lot of these systems combine different techniques and also perform both of the detection tasks, and thus, their classification is considerably complex.

4.3.2.2 Anomaly Detection

An anomaly can be defined as something that deviates from what is standard, normal, or expected. This way, the first step to build an anomaly detector is to establish what is considered the normal behaviour of a system (users, networks, audit trails, system calls, etc.). Once this is defined, the deviations from the normal behaviour that the system detects will be classified as suspicious or intrusive.

Anomaly detection depends greatly on the supposition that users and networks behave in a sufficiently regular way and therefore, any significant deviation from such behaviour could be considered as an evidence of an intrusion.

The great advantage of anomaly detection is that the system is able to learn the studied object's normal behaviour and from that point detect deviations classifying them as intrusions. This way, it is demonstrated that these systems are capable to detect unknown attacks.

On the other hand, by definition they can only detect unusual behaviours, but these are not necessarily illicit. Therefore, one of the biggest problems of this kind of IDSs is the high rate of false positives. Another disadvantage is the lack of clarity of the process; it is a fuzzy process. A patient intruder could work slowly and act cautiously in order to modify the profile of the users and make his own actions become acceptable for the IDS not generating any alert as they should (false negatives). In other situations, it is not enough to generate an alert but an explanation of what had happened is as important as the alarm itself. The nature and reasons of the anomalous behaviour should be explained.

Similarly to misuse detection, there are several ways to implement methods for anomaly detection. Heuristic and statistical mechanisms are used in order to be able to adapt to the changes of the studied object and also to detect unpredictable changes. There are also other approximations that try to include other techniques for such function.

1. **Knowledge Based Systems**

   Expert Systems. At the beginning, these were the most used systems for IDS. As mentioned earlier, the IDES model was the first expert system developed for intrusion detection. The model proposed by Denning is based on the hypothesis that security violations can be detected by monitoring the audit trails of systems. It looks for abnormal use patterns and uses rules to acquire knowledge from audit trails.

Later, the SRI put some effort enriching, optimizing and redesigning the IDES prototype and the NIDES (Next-generation Intrusion Detection Expert System) system was born. NIDES runs on its own work station, called NIDES host, and analyses audit data gathered from several interconnected systems looking for activities that could indicate unusual and/or malicious behaviours of users. Two complementary detection units carry out the analysis: the signature analysis subsystem based on rules (using the P-BEST tool) and the statistical anomaly detection subsystem based on generated profiles [34]. Already mentioned, there is another expert system called NADIR, which works analysing the network activity.

2. Statistical Methods Based Systems

As explained earlier, both IDES and NIDES rely on statistical analysis-based expert systems. They include profiles to represent the behaviour of subjects with respect to objects in terms of metrics and statistical models. They also use rules for the acquisition of knowledge from audit trails and also for the detection of anomalous behaviour. One of the basic components of such systems are the activity profiles. This component characterizes the behaviour of a subject (usually users) with respect to an object (files, programs, logs, terminals, etc.). This characterization is made by establishing metrics and statistical models (like operational model, significance model and standard deviation, multivariate model, Marcov process model and temporal series model). Other statistical models have also been used to create IDSs: Finite mixture model [53], measures based on X^2 [54] or the statistical technique called Camberra [55]. The same authors have developed a system called ISA-IDS (Information and System Assurance Laboratory Intrusion Detection System) [56] that uses X^2 to detect anomalies and decision trees for misuse detection. DuMouchel and Schonlau test statistical methods to obtain user profiles [57]. Finally, Mei-Ling Shyu et al. presented a work based on the Principal Component Classifier (PCC) technique [58].

3. Machine Learning Based Systems

This kind of systems have been and still are the most studied ones as methods to model normal behaviours. Due to the great variety of machine learning models, a lot of research have been carried out trying to find the most suitable in terms of detection accuracy, reduction of false positives and required computational time. There are works that have treated several models at the same time. Sometimes to make comparisons between different models and other times to choose the most suitable model according to the type of attack to be treated. Reference works in the area of systems based on machine learning for anomaly detection have been carried out by Barbará et al. [59] in George Mason University. Their project called ADAM (Audit Data Analysis and Mining) uses incremental data mining techniques (Bayes estimators) to detect anomalous traffic patterns in real time. Another major project is MINDS (Minnesota Intrusion Detection System) [60] developed in the University of Minnesota. MINDS is a complement to a misuse detection IDS (Snort) and detects anomalies based on the SNN algorithm (Shared Nearest Neighbour). Once the anomalies are detected, a summarization of them is carried out by the analysis of the pattern associations from where characteristics that determine an attack are obtained and are added to the knowledge base as new signatures. Some other significant projects are: ADMIT [61] and the PhD thesis by T. Lane from Purdue University [62]. This area of research has been very productive in last years.

4.3.3 Response

Most of IDSs trigger a basic response method when they detect an attack: notification. This kind of response is passive and its only aim is to inform the administrator about the occurrence of an attack. There are several ways to warn the administrator and common ones are: on-screen message, email, SMS, etc.

During the last years though, automatic response to attacks have been considered and have gain much popularity. This is known as active response or automatic response.

Active response is an emerging research area. Current response methods ignore the real cost of an intrusion and active response may help to minimize them. Nevertheless, in order to become an efficient option, it is going to be necessary to develop more precise sensors as the main problem with automatic responses is what happens when the alarm is not real (false positive). Responses to false alarms are unnecessary and can be expensive too [63]. Even worse, they can cause a degradation of the service provided to legitimate users of the system that could become a denial of service in the worst case. This subject is explained in more detail in [64], [65], [66] and [67].

Another problem is the way that current IDS vendors use the terms related to automatic response. In an effort to differentiate their own product from competitors, they include supposed features that confuse the final users. A clear example of this is the term Intrusion Prevention System, IPS. In most of the cases they are not a new product but the same old IDS installed in-line plus some firewall features to be able to block some connections and perform some other response actions.

4.4 Intrusion Detection Systems in Wireless Sensor Networks

In the last years research on providing security to WSN has focused mainly in three categories [68]: key management, authentication and secure routing, and secure services. Key management consist in establishing cryptographic keys between nodes to enable encryption and authentication. In authentication and secure routing category, several protocols have been proposed to protect information from being revealed to an unauthorized party and guarantee its integral delivery to the base station. In the third category, secure services, there has been a slight progress in providing specialized secure services, like secure localization, aggregation and time synchronization.

[69] says that wireless sensor networks are vulnerable to adversaries as they are frequently deployed in open and unattended environments. As a solution, preventive mechanisms can be applied to protect them from an assortment of attacks. However, more sophisticated methods, like intrusion detection systems, are needed to achieve a more autonomic and complete defensive mechanism, even against attacks that have not been anticipated in advance.

As explained in [70], intrusion detection has received some attention in wireless sensor networks before. Most work has focused on local detection, i.e., allowing nodes to locally detect specific attacks which are performed in their neighbourhood.

Several intrusion prevention techniques have been introduced for sensor networks over the last few years [71], [72]. Such prevention measures, like encryption and authentication, can be used to reduce intrusions but cannot eliminate them. For example, encryption and authentication cannot defend against compromised sensor nodes which carry the private keys. From the experiences of security research, no matter how many intrusion prevention messages are inserted in a network, there are always some weak links that one could exploit to break in. Due to this reason, an adversary will go unnoticed and this is

likely to lead to failures in the normal operation of the network. If the intruder is detected soon enough, we can take any appropriate measures before any damage is done or any data is compromised [73].

As it is described in [16], an IDS in a wireless network sensor must: work with localized and partial audit data because of the lack of centralized points; use a small amount of resources; assume that no node can be fully trusted; be fully distributed; and be able to withstand an attack against itself.

Da Silva et al. [71] and Onat and Miri [74] propose similar IDS systems, where certain monitor nodes in the network are responsible for monitoring their neighbours. They listen to messages in their radio range and store in a buffer specific message fields that might be useful to an IDS system running within a sensor node. Kargl et al. [75] focus on the detection of selfish nodes that try to preserve their resources at the expense of others. Loo et al. [76] and Bhuse and Gupta [77] describe two more IDSs for sensor networks. Both papers assume that routing protocols for ad hoc networks can also be applied to WSNs. In all the above work, there is no collaboration among the sensor nodes.

In [73], a intrusion detection system is proposed, based on a distributed intelligent agent-based system. It performs intrusion detection in a fully distributed manner. Each node is loaded with an independent IDS agent, capable of detecting intrusions locally based on the data collected by itself and by other neighbouring nodes. Responses and actions taken to isolate these intrusions are based on collaborative decisions made by the set of participating nodes.

The few, if not only, collaborative approaches we are aware of focus on the local detection of selective forwarding attacks [68] and sinkhole attacks [78]. More extensive work has been done in intrusion detection for ad hoc networks [79]. In such networks, distributed and cooperative IDS architectures are also preferable. Detailed distributed designs, actual detection techniques and their performance have been studied in more depth. While also being ad hoc networks, wireless sensor networks are much more resource constrained. We are unaware of any work that has investigated the issue of intrusion detection in a general collaborative way for wireless sensor networks. [70] itself approaches a more oriented to collaboration between sensor than specific attacks detection intrusion detection system.

[69] presents as a solution a lightweight intrusion detection system, called LIDeA, designed for wireless sensor networks. This IDS is based on a distributed architecture, in which nodes overhear their neighbouring nodes and collaborate with each other in order to successfully detect an intrusion. LIDeA uses components and interfaces of TinyOS, a free and open source component-based operating system and platform targeting wireless sensor networks.

Hybrid Intrusion Detection Systems (HIDS), based on hybrid star architecture, have also been proposed. In [80], in fact, an architecture of HIDS applied to Cluster Wireless Sensor Networks (CWSN) is presented, detecting intrusions by Cluster Heads (CH). The proposed HIDS consists of an anomaly detection model and a misuse detection model. It filters a large number of packet records, using the anomaly detection model, and performs a second detection with the misuse detection model, when the packet is determined to intrusion. Therefore, it efficiently detects intrusion, and avoids the resource waste. Finally, it integrates the outputs of the anomaly detection and misuse detection models with a decision making model. This determines the presence of an intrusion, and classifies the type of attack. The output of the decision making model is then reported to an administrator for follow-up work. This method not only decreases the threat of attack in the system, but also helps the user handle and correct the system further with hybrid detection.

Another HIDS is presented in [81], which is based in anomaly and misuse technique. The attack detections are achieved through the collaborative use of global agent and local agent integrated in

application layer of sensor node. It is proposed a defence method and four algorithms to detect and isolate the malicious node from the network.

4.4.1 Proposed IDS for WSN

All of the mentioned attacks can be detected using an Intrusion Detection Systems. In one hand, IDS can be classified considering the detection method used by the system. Among all these kinds of security systems, anomaly detection systems can detect anomalous behaviour in the network that can be considered as intrusions or attacks. Misuse detection systems, instead, can detect data patterns that match with previously identified attacks. Otherwise there are more than the two mentioned IDS classes. More IDS information can be found in [21].

On the other hand, there are another two main classes to categorize IDSs: centralized and distributed. Centralized network intrusion detection systems are characterize by distributed audit collection and centralized analysis. In other words, an IDS agent running on a host (usually the gateway, which is connected with the administration part) is fed by sensor nodes. In this case, the IDS agent analyses the data and possibly detects on-going attacks. When a routing attack is performed control packets can be prevented to reach the IDS agent so it could get an erroneous view of the network, resulting in failing to detect the attack. This scenario and the possibility of the failure of the IDS agent are the worst cases in centralized IDS.

The distributed solutions, instead, is based on the detection logic that the sensor nodes have. Potentially, these kind of systems are more resilient to network level attacks, since it is still possible to detect the attacks locally, even in the case when the network infrastructure is damage (although in this case there is a risk of the IDS system getting to a wrong decision, due to a non consistent view of the global status of the network). Additionally, distributed solutions may need the execution of agreement protocols to allow each node to share its local view of the network with a set of neighbours. The consumption or resources in this case increases, due to an increase in the number of transmissions. Referring to the collaboration between nodes in a WSN, it must be taken into account that attacks could perfectly come from the inside of the network (a compromised sensor node with legitimate access to the network may launch the attack). This idea is presented in [96] and concludes in the fact that no node can be trustful. In that work it is also explained that an adversary can physically capture a sensor node form a network (sensor nodes are deployed in a certain area) and reprogram its ROM to change its behaviour. To identify these possible situations, they propose a Local Intrusion Detection Component that analyses local features in the nodes to detect whether its host is suffering attacks from other malicious nodes. The same idea of protection against malicious sensors or inside attackers is presented in [97], where they recommend a solid malicious sensor detection algorithm to be robust and fault-tolerant. They continue saying that detection of insider attackers may be accomplished by exploring the correlation among neighbouring nodes. In a typical scenario sensors are expected to have almost constant communication and computation workloads in close proximity. This constant behaviour does not match with the potential adversary, who would misbehave in some aspects with respect to normal nodes, such as broadcasting or dropping excessive packets, generating abnormal data packages. This node behaviour deviated remarkably from a typical application-specific range and can be considered as a faulty or malicious node. As a solutions [97] propose a localized algorithm for insider attackers detection inspired from the spatial correlation existent in the neighbourhood.

As the main conclusion, although may result in a increase in the resources of a sensor node, the global security level that gives a distributed solutions to the network is considered more reliable than the one that centralized one. As it is described in [96], the centralized architecture is not suitable for an intrusion detection system to detect as many types of attacks as possible, due to the low data rate of wireless

communication and limited energy of the sensor nodes could not afford to pass the massive audit data to a base station to be analysed. The centralized solution performance does not guarantee the correct intrusion detection in the network due to the threat of being unable to detect those intrusions in certain situations. However, in a distributed intrusion detection system no nodes are trustful, due to inside attackers. For that reason is necessary to propose an agent able to detect anomalies in its host neighbours. The protection of the nodes is also necessary so it is high recommended to implement a local agent for the nodes able to analyse possible local features changes.

Risk	Solution
Stealing Data (Confidentiality)	Encryption methods
Altering/Generating False Data (Data Integrity)	Prevention Methods (IDS)
Attacks on Service Availability	Prevention Methods (Access Lists, Firewall)
Denial of Sleep Attacks (Energy)	Prevention Methods (IDS)
Routing Attacks (In general)	Distributed IDS

Table 4-1: Risk/Solution Summary Table

The proposed IDS is composed by an distributed architecture and implemented through hybrid anomaly detection system. In this system every node runs a detection system, which is in charge of identifying suspicious nodes that are near them. These suspicious nodes are inserted temporarily in a blacklist and an alarm is sent to the central agent. The central node gets the information of the rest of nodes, and in case of a false alarm this central node will send a message of false positive to the first node to erase the positive node from blacklist. Instead, if it is a true alarm, the central node will report to rest of nodes to put in the blacklist the suspicious node.

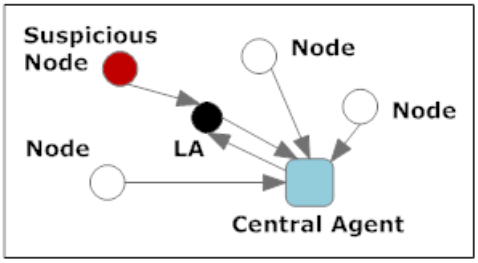

Figure 4-8: IDS architecture.

This solution combines misuse and anomaly based techniques in a distributed hierarchy for improving resilience and performance. The following agents can be found in the detection engine:

- IDS Local Agent (LA), that is in charge of analyse the traffic and gather the data to send to the IDS Central Agent.
- IDS Central Agent (CA), that is in charge of verify the received data and detect possible attacks.

5 Cognitive node architecture for the wireless-radio environment

The employment of sophisticated tools for data analysis in distributed or structurally complex systems requires the development of specific data fusion strategies to integrate the heterogeneous information coming from the environmental sensors. In such a framework, intelligence distribution is one of the most interesting research fields: the logical tasks are partitioned in real time between the various architecture components: intelligent sensors, intermediate nodes and remote control centers. Typical tasks such as context analysis and recognition are decomposed into hierarchical chains of subtasks. Each logical block of such functional chains receives as inputs the data produced by the lower block and produces a representation of the environment at a higher abstraction level. The latter will be supplied to the higher level blocks, and so on, obeying strict temporal constraints.

The data fusion process can therefore be sequentially assigned to different levels of the architecture in a distributed way, in order to output an overall representation of the environment and specific indications of situations of interest.

A typical description of a security system can be done in terms of a hierarchical tree structure, where sensors, elaboration nodes and remote control centres are connected through heterogeneous communication channels. Within such a structure each sensor contributes to the global monitoring by gathering specific data. Since sensors ore presently provided with (narrow) elaboration skills, raw environmental data are locally analysed and aggregated metadata are sent to the intermediate elaboration nodes.

Control centres are spots of the architecture where all relevant environmental data are conveyed by the intermediate elaboration nodes and gathered in real time in order to be usable (possibly by human operators by means of specific interfaces) in order to face out of the ordinary situations with targeted actions.

In this report the role of elaboration nodes in such architectures is analysed. Advantages (in implementation and application) deriving from the use of biologically inspired cognitive models are pointed out. The application of ambient intelligence to pSHIELD is eventually depicted.

5.1 Intelligent systems

Intelligent systems are defined as such [104] whenever they are designed to integrate the Environmental Intelligence Paradigm [103] with the traditional security applications. The Paradigm defines as fundamental properties of “intelligent” systems the capacity of context analysis and intelligent distribution.

5.1.1 Context analysis

Many recent works have been having as main objective the realization of tools for the automatic analysis of the context; such analysis is usually focused on the recognition of the behaviour of the people in the scene, since the ability to classify behavioural information is of fundamental importance in managing security and in preventing critical situations in risky environments. Some examples of Ambient

Intelligence applications to security systems are: classification of interpersonal and person-to-object interactions, threat recognition [104][102][100].

The utilization of movement schemes for behavioural analysis and anomaly recognition is an effective approach, as it allows to accurately establishing the movement of various entities within the environment. Trajectories are grouped by means of specific grouping techniques; appropriate behavioural models are hence constructed.

A model for different human activities must be constructed taking into account the natural variability of human behaviour. Each person performs the very same activities in a different way. Moreover, some actions can acquire different meanings depending on the overall global situation. Therefore, being not realistic to model human behaviour deterministically, appropriate probabilistic models for the description of activities and interaction are often used.

5.1.2 Distributed intelligence

Distributed intelligence, a distinguishing feature of intelligent systems, is a crucial factor for concurrent optimization of the communication channel between the blocks of the architecture and of the global elaboration skills of the systems. The possibility of data analysis at low levels in the architecture implies, for instance, less data load at higher levels and denies overloads or delays that could easily occur in case all the elaboration was concentrated in a single spot in the architecture. Such a solution also provides more robustness by means of delocalization and redundancy of the elaboration activity: distributed systems are, as a matter of fact, less susceptible to single components breakdowns.

Typical tasks can be decomposed in a chain of logical modules, organized in a hierarchical structure: low level modules produce as outputs the meta-data needed by the higher level modules. This way, starting from raw data (non-processed data), a representation of the environment at a higher abstraction level is obtained at each level of the architecture. Such decomposition, originally proposed in [101] defines the intelligence distribution paradigm in terms of logical modules allocation within the different physical elements of the architecture, with autonomous data elaboration abilities.

The modularity of the functionalities of intelligent systems and their allocation in different subsystems must however guarantee a quality in the analysis, which must at least be equal to the case where the elaboration is located entirely in one only architecture block. It is therefore necessary for the modules to communicate to each other, independent of their physical location and the link between them (e.g. wireless or wired). Moreover, modules distribution and the necessity of saving the data generated from them, makes it necessary to memorize representations of detected events in suitable structure bounds to the physical device.

There are three typologies of modules, defined at different abstraction levels: representation modules (information's elaboration tasks: the output is a higher level symbolic representation of the data than the input), recognition modules (algorithms compare input data with a set of models) and communication modules (which produce a codified representation of the input data, suitable for their transmission).

Such modules are the logical components by which collect together the different functionalities, namely the parameterization alphabet of the applicative middleware for the security. This allows the architecture to work in a dynamic, reliable and flexible way. The chains of modules can be loaded by means of the functions for the dynamical management of the network resources, in the control centers or when activating the functionalities requested by the user, or whenever changes in the state of the network occur.

5.2 Cognitive systems

Two limitations of the intermediate elaboration modules in an intelligent system are their passivity and the inability of learning based on experience.

In fact, despite the development of specific applications capable of semantic context analysis, such systems are passive, since they are not designed to work over the environment to solve threat situations. Usually, the chain of modules of context analysis located within the elaboration nodes, produces, in case of anomalies, specific alarms which are sent to human operators for decision making and action.

Cognitive systems can overcome these limitations by means of a cognitive cycle (sensing-analysis-decision-action). Cognitive systems indeed implement a model which imitates the brain functionalities and not only are able to correctly analyse the meaning different situation, but can also to act consequently after a decision. A cognitive system has the capability of interacting in a closed cycle with the outside world by means of the actuators present in the environment.

The cognitive system has an internal model which describes the actuators related to itself and the action they can make towards the environment (embodied cognition).

Cognitive systems make use of a learning phase to codify within appropriate data structures the behavioural models, based on experience. To be precise, the information stored is the one concerning the relations between changings in the state of the system and changings in the outside world (and vice-versa). This way, a cognitive system can recognize some situations and forecast, through an inference mechanism, their future development without any information on rules.

A cognitive system can also learn from experience the decisional models of a human operator, based on his actions as a reaction to specific environmental situations. The knowledge acquired is used to model specific automatic decision routines based on context meta-data coming from the chain of logical blocks of analysis. One can therefore define automatic decision blocks at different abstraction levels based on the information concerning the state of the system, the current events, the predicted events and the classification of the current scenario.

A cognitive system than overcomes the typical limitations of simple intelligent systems by adding to the architecture of the system appropriate logical blocks devoted to decision and learning.

5.3 Cognitive model

Cognitive systems are based on a neurophysiological model of reasoning and awareness [99]. In this model, a cognitive entity is described as a complex system which is able to learn incrementally – on the basis of experience – relations between themselves and the external world. Neuroscientific conceptualization of cerebral human functions defines two specific devices, called proto-self and proto-core, which are devoted to the monitoring and management of the internal state of the entity and of the external world respectively. The possibility of gaining access to its own internal state (self-consciousness) is for the cognitive entity as necessary as the ability of analysing the environment. According to this model the sensors available to a cognitive entity can be divided into endo-sensors (or proto sensors) and eso-sensor (or core sensors) depending on whether they are used for internal or external states monitoring.

The behaviour of a cognitive entity interacting with the world is described by the cognitive cycle and can be divided (Figure 1) in four fundamental steps. Sensing, Analysis, Decision, Action. These steps represent, as time flows, an infinite sequence, since the state (internal and external) which is perceived at each step is (directly or indirectly) influenced by past Actions made by the cognitive entity itself.

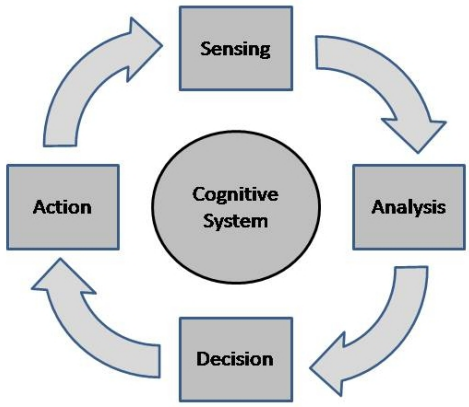

Figure 5-1 - Cognitive cycle

Therefore, the conceptual architecture of a cognitive entity is made of four logical blocks (Figure 5-1):

Sensing: a cognitive system constantly gets information about the core- and self-states by means of endo- ad eso-sensors.

Analysis: the data coming from the sensors are fused in order to obtain a common description of the external world as well as the internal state of the cognitive system. Input data are than analysed to detect events, which can in turn be either proto events (εP), relative to significant changes in the internal state of the system or core (εC), relative to changes in the external world. From such data, a cognitive entity is able to create a model of probability distributions of proto and core events, $p(\varepsilon_t^P | \varepsilon_t^P)$ and $p(\varepsilon_t^C | \varepsilon_t^C)$. This model (first order neural pattern) does not account for possible interactions between core and proto events and can be regarded as a couple of Dynamic Bayesian Networks (proto-DBN and core-DBN).

Decision: according to the experience of the cognitive system (obtained through a codification of past events filtered through appropriate data structures) and to the analysis of the current internal and external states X_P and X_C, the system selects the most appropriate strategy ST in order to get the desired configuration of the system $\{X_P, X_C\}$. The target configurations $\{X_P, X_C\}$ are selected in order to get stability (homeostasis) with respect to specific behavioral models (learned or available).

Action: this module implements the active interaction of the system towards the surrounding environment: an appropriate action a_t is selected based on the strategy ST chosen during the previous step. Such an action is executed on the environment or on the system itself by means of suitable specific actuators.

5.4 The simulator

The PSHIELD simulator has been developed in the cognitive framework described above.

5.4.1 Scenario

The scenario consist in a number of entities (agents) carrying a mobile device which is able to transmit and receive data at 3 different frequencies (namely 900, 1800 and 1900 MHz) to a centralized control centre. The agents move randomly throughout a radio-disturbed environment, where randomly placed jammers emit a disturbing signal. The jammers can be either fixed or moving and their emitted signal follows the Rayleigh distribution with fixed parameters. Fixed jammers positions and characteristics are stored in an XML file, which is loaded in the setup stage together with the map of the ground. A scenario with 2 moving agents transmitting at a frequency of 1800 MHz and one Jammer, with their respective radii of sensing and influence is depicted in Figure 5-2.

The mobile devices periodically send a single radio data to the control centre, where a running cognitive node receives and elaborates it. Also, a periodical polling is performed by the agents to question the node, which answers back.

Figure 5-2 - Scenario with two agents and one fixed jammer

A radio data sent by an agent contains the following pieces of information:

- Position of the agent (x,y) on the mapped ground: this is generated by a trajectories simulator. It simulates a GPS sensor on the mobile device. If a video monitoring of the ground area is

available, positioning data coming from a tracker can be possibly fused to GPS data to obtain a better position estimation.

- Frequency of transmission: this can be chosen among the three available frequencies at the beginning of the simulation.
- Power of the transmitted signal: fixed.
- Power of the signal received from the node: this depends on the distance and it is calculated through FSPL. Also, it can be disturbed by jammers.
- Possibly detected jammers' estimated power: each jammer has a typical radius (coded in the XML configuration file) of influence, inside which the agent can measure its power.
- ID of possible neighbour agents (within a fixed sensing radius).

A slightly different scenario can be also set by introducing a moving jammer: an agent carrying a jamming device can be introduced in the scene. Such an intruder-agent differs from thee others as he obviously disturbs communications to the node. Also, he communicates a false GPS survey to the node. A scenario with 2 moving agents transmitting at a frequency of 1800 MHz and one Jammer, with their respective radii of sensing and influence is depicted in Figure 5-3.

Figure 5-3 - Scenario with moving jammer (intruder)

5.4.2 Cognitive model application

The radio data reception represents, from the node point of view, the sensing logical block of the cognitive cycle. The agents' mobile terminals are the sensors which monitor the environment sending a radio survey (radio sensors) and a positioning piece of information (GPS sensor).

The node then analyses all the data received form each agent, both singularly and collectively. For each agent, the signal-to-noise and distortion ratio (SINAD) of the received data packet is computed. Also, the relative positions the agents are compared, on the basis of the datum sent by an agent himself and of the fused data sent by the agents in the sensing range. By means of a voting algorithm, rankings are assigned to the IDs of each agent. The intruder's position and ID are worked out as soon as enough information is gathered, based on such rankings.

In the decision stage, the SINAD datum is compared to an acceptable (fixed to 10 dB) threshold. If the communication with an agent turns out to be too disturbed, a suitable strategy ST is chosen to schedule a change in frequency transmission.

The action block provides a change in the state of the system. As already explained, this module implements the active interaction of the system towards the surrounding environment or towards itself: the action of changing frequency is selected based on the strategy ST chosen during the previous step. Such an action is executed on the system itself by means of suitable actuators, namely the agents. Actually, as already pointed out, through a periodical polling, the agents themselves ask the node for information: however this does not change the heart of the matter.

The detection of the intruder does not trigger a decision and a subsequent action in the cognitive cycle. The information relative to the false agent is simply communicated to an interface. Such an interface could simply be in a control centre, or could display data on the mobile devices, thus leaving the decision step under human control. Alternatively, a strategy could be implemented to be learned by the cognitive node in a future perspective.

6 Conclusions

Section Cryptography Framework presented different key management and distribution protocols. Key management is an integral part of most cryptographic systems. Different key management schemes where discussed including conventional and more advanced ones. An important category is the key pre-distribution scheme which includes several probabilistic, polynomial and domain knowledge enhanced schemes. It is concerned with the a priori distribution of the keys on network constituents prior to system deployment. Also, for systems requiring more frequent rekeying and key management operations online, the dynamic key management category has emerged. Architecture specific schemes were also discussed given the nature of the pSHIELD's application scenario, i.e., hierarchical heterogeneous network. The common concerns in selecting a key management scheme for a given system are typically the kind of attacks to protect against, scalability, node storage capacity, the extent of resilience against node capture and the ability of self-healing upon an attack. There is no single answer as to which scheme is the best; however, as far as pSHIELD's application scenario is concerned a customised hierarchical scheme could be beneficial. Authentication within that scheme should not be resource demanding as detailed in Section 3.2.1.

This document also presented a range of security issues in networked embedded systems, typically lightweight devices, and discussed the intrusion detection system in wireless sensor networks. A wireless sensor network communication method study is approached, analysing different communication methods. Three different architectures were discussed in the wireless sensor networks topic, i.e., star, mesh and hybrid architectures. Intrusion detection system classification was also approached, showing the different intrusion detection methods used to classify them: source classification, analysis strategy classification and response classification.

With respect to the network architecture in wireless sensor networks, it basically depends on the energy consumption capacity of the nodes. Star architecture provides simplicity and low energy consumption because all the nodes connect and send information to a unique central node. The main problem of this structure is that if the central node falls, all the network comes down. A simple solution is the idea of mesh architecture, every node send and receive information from every single node in the network. However this structure makes the nodes consume more energy, reducing the node’s battery lifetime. The hybrid architecture, a combination of star and mesh architectures, would confere the network a good combination of power consumption, robustness and versatility. The idea of having some central nodes that share the information with some other nodes gives the chances to create an efficient and well manageable wireless sensor network where the central nodes send and receive information from other central nodes. It could be described as a network where there exist some clusters that share information between them through central nodes, and central nodes (which usually have more power and computing resources) form a mesh among them..

In terms of communications IEE802.11.x, IEE802.15.4 and ZigBee are highlighted in the study. The first one, IEE802.11.x, has an interesting feature: its relatively high bandwidth data transfer between computers or other devices. But due to the power requirements it is normally not used in wireless sensor networks. However, this option could be very useful in local area networking, where more powerful nodes could be relatively accessible to the management central. IEE802.15.4, a method that works in 2.4GHz band frequency, is a recommended communication system in wireless sensor networks as it was designed for wireless sensing requirements, achieving a low energy consumption level, an essential fact in this kind of networks. ZigBee is an extension of IEE802.15.4 standard and is

seeking to standardize higher level applications. There are other communication options, such as Bluetooth, but have some limitations in the protocols that may end in communication problems. Taking all this into account, it is considered that IEE802.15.4, which includes ZigBee, is the most appropriate communications option for wireless sensor networks.

Three intrusion detection systems types are approached, explaining their main characteristics. Information source class is the first one mentioned, explaining the three different information sources that this class is concerned with: hosts, networks and applications. Analysis Strategies is the second class of IDS mentionned, explaining the Misuse Detection, based in attack models that are known in advance, and Anomaly Detection, based in events that are not part of the normal or common behaviours. Response IDS, the third and the last class approached, is based on the response, active or passive, that the IDS is designed to give to a detected attack or event. The active response is an emerging method that has been studied during the last years. One of the most important requirements when designing an IDS tailored for embedded systems is that the system itself must be lightweight, due to the necessity of low energy consumption.

One of the characteristics in the pSHIELD project is reaching a generic level in the implementation of IDS in wireless sensor networks. This reason is the key to understand that, in this project, there would be no point in specifying an ideal communication protocol or network structure, the intrusion detection system itself should be adapted to the network and protocol where it is deployed. In any case, the intrusion detection system class to be developed is considered to be the Anomaly Detection system. In this way, the system would base its knowledge on previous network activities that are considered to be regular behaviour of the network. A Misuse Detection system is also a viable option, but this supposes we know beforehand every single attack or intrusion class, something very difficult or almost impossible to specify in a generic system oriented to any kind of wireless sensor networks, or that we are able to update the attack knowledge base, which in turn means we need to use additional transmissions (wasting more power in the node).

As the main conclusion for the IDS system, although it may result in a increase in the resources of a sensor node, the global security level that is achieved using a distributed solution for the network is considered more reliable than the one using a centralized one. As it is described in [96], the centralized architecture is not suitable for an intrusion detection system to detect as many types of attacks as possible. Due to the low data rate of wireless communication and limited energy of the sensor nodes they could not afford to pass the massive audit data to a base station to be analysed. Thus the centralized solution performance does not guarantee a correct intrusion detection in the network due to the risk of being unable to detect those intrusions in certain situations. However, in a distributed intrusion detection system no nodes are trustful, due to inside attackers. For that reason it is necessary to propose that an agent running on every node be able to detect anomalies in its neighbour nodes. The protection of the nodes is also necessary so it is high recommended to implement a local agent for the nodes able to analyse possible local features changes.

According to T4.1 and T4.2 objectives new technologies enabling smart SPD driven transmissions have been proposed. In particular, the cognitive radio (CR) paradigm, which is usually based on Software Defined Radio (SDR), has been proposed to deal with such transmissions. CR is composable and expandable and modular by definition. In fact, it has been designed to accommodate these features.

The implemented Cognitive Radio Node is able to receive radio parameters from moving hosts and automatically detect possible threats. The internal architecture of the Node learns typical safe environments features thus detecting the presence of external attackers by analysing radio parameters.

In a considered scenario, the cognitive node always updates the radio parameters (SNR, BER and Transmitter Power, PTX) for the self-awareness purposes. There are some specific provisions considered to design this kind of simulator used for the Security, Privacy and Dependability (SPD) in the context of integrated and interoperating heterogeneous applications.

When an agent enters the scene, the cognitive node becomes aware of the radio parameters of the agent either by using the spectrum sensing technique or from a direct communication from the agent itself. In this way the node can update its radio information for using the radio resources efficiently and securely. The cognitive node has an internal knowledge of all the radio parameters which would be considered in the selected environment and their respective variation models. The node knows itself from a configuration database what frequencies are used by which agent and which frequencies are free to use. If a new agent enters in the scene while continuing communication, the cognitive node sense the radio parameters of the agent and is able to modify and adapt agent's radio parameters when necessary.

In the presence of a jammer of specific frequency in a cluster, the cognitive node sends a message to the agents to adjust the radio parameters properly, i.e., by changing either the frequency or the transmission power (spread spectrum or noise based data transmission of signals).

Moving agents in the scene and the presence of jammers are dynamically created through a specific simulator that was built to this aim. The simulator sends to the cognitive node is the positioning data, namely the trajectories of the agents (like a tracker) and radio data on the situation. More specifically, each agent is controlled by the cognitive mobile node, considered as an entity, after the registration process in the area under observation, periodically sends information on the quality of communication.

7 Dissemination activities

In order to disseminate the results achieved during project-related activities, partners have participated to international conferences and forums where part of the work performed in pSHIELD has been discussed. pSHIELD related publications are:

- L. Bixio, M. Ottonello, M. Raffetto, and C.S. Regazzoni, “Comparison among Cognitive Radio Architectures for Spectrum Sensing,” EURASIP Journal on Wireless Communications and Networking, vol. 2011, Article ID 749891, 18 pages, 2011. doi:10.1155/2011/749891

- L. Bixio, L. Ciardelli, M. Ottonello, M. Raffetto, C. S. Regazzoni, Sk. S. Alam and C. Armani, “A Transmit Beamforming Technique for MIMO Cognitive Radios,”, Wireless Innovation Forum Conference on Communications Technologies and Software Defined Radio, SDR'11 - WInnComm - Europe, Brussels, Belgium, June 22-24, 201

- S. S. Alam, L. Marcenaro and C. Regazzoni, “Opportunistic Spectrum Sensing and Transmissions”, submitted for publication

- Iñaki Garitano, Roberto Uribeetxeberria and Urko Zurutuza, “Review of SCADA Anomaly Detection Systems”, Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011, Salamanca (Spain) in April, 2011, ISBN 9783642196447

- Urko Zurutuza , Enaitz Ezpeleta, Álvaro Herrero and Emilio Corchado “Visualization of Misuse-based Intrusion Detection: Application to Honeynet Data”, Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011, Salamanca (Spain) in April, 2011, ISBN 9783642196447

- Ekhiotz Jon Vergara, Simin Nadjm-Tehrani, Mikael Asplund and Urko Zurutuza, “Resource Footprint of a Manycast Protocol Implementation on Multiple Mobile Platforms”, Fifth International Conference on Next Generation Mobile Applications, Services and Technologies,NGMAST 2011, Cardiff, Wales, UK, 14-16 September 2011.

8 References

[1] Department of Energy of USA, "Smart Grid. Department of Energy," vol. 2010.

[2] S. Ravi, A. Raghunathan, P. Kocher and S. Hattangady, "Security in embedded systems: Design challenges," *ACM Transactions on Embedded Computing Systems (TECS),* vol. 3, pp. 461-491, 2004.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks,* vol. 38, pp. 393-422, 2002.

[4] E. A. Lee, "Cyber physical systems: Design challenges," in *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on,* 2008, pp. 363-369.

[5] Zigbee Aliance, "Zigbee Aliance," vol. 2010, .

[6] International Society of Automation, "ISA100, Wireless Systems for Automation," vol. 2010, .

[7] HART Communication Foundation, "HART Communication Protocol and Foundation," vol. 2010, .

[8] J. Paek, B. Greenstein, O. Gnawali, K. Y. Jang, A. Joki, M. Vieira, J. Hicks, D. Estrin, R. Govindan and E. Kohler, "The tenet architecture for tiered sensor networks," *ACM Transactions on Sensor Networks (TOSN),* vol. 6, pp. 1-44, 2010.

[9] A. A. Cárdenas, S. Amin and S. Sastry, "Research challenges for the security of control systems," in *Proceedings of the 3rd Conference on Hot Topics in Security,* 2008, pp. 1-6.

[10] D. Dolev and A. C. Yao, "On the security of public key protocols," in *22nd Annual Symposium on Foundations of Computer Science,* 1981, pp. 350-7.

[11] D. L. Evans, K. H. Brown and W. M. Director, "Security Requirements For Cryptographic Modules," *Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD,* 2001.

[12] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards," in *Proceedings of the 18th Conference on USENIX Security Symposium,* 2009, pp. 1-16.

[13] M. M. Sebring, E. Shellhouse, E. M. Hanna and A. Whitehurst, "Expert systems in intrusion detection: A case study," in *11th National Computer Security Conference: Proceedings, 17-20 October, 1988,* 1988, pp. 74.

[14] J. S. Wilson, "Wireless sensor networks: Principles and applications," in *Sensor Technology Handbook*Anonymous Newnes, 2005, .

[15] D. J. Cook and S. K. Das, *Smart Environments: Technologies, Protocols, and Applications.* Wiley-Interscience, 2005.

[16] A. STETSKO, "INTRUSION DETECTION FOR WIRELESS SENSOR NETWORKS," .

[17] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad Hoc Networks,* vol. 1, pp. 293-315, 2003.

[18] J. Hill and D. Culler, "A wireless embedded sensor architecture for system-level optimization," 2001.

[19] C. Karlof, N. Sastry and D. Wagner, "TinySec: A link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems,* 2004, pp. 162-175.

[20] P. Albers, O. Camp, J. M. Percher, B. Jouga, L. Mé and R. Puttini, "Security in ad hoc networks: A general intrusion detection architecture enhancing trust based approaches," in *Proceedings of the 1st International Workshop on Wireless Information Systems (WIS-2002),* 2002, pp. 1-12.

[21] U. Zurutuza, "Data mining approaches for analysis of worm activity toward automatic signature generation," 2008.

[22] H. Debar, M. Dacier and A. Wespi, "A revised taxonomy for intrusion-detection systems," *Annals of Telecommunications,* vol. 55, pp. 361-378, 2000.

[23] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Chalmers Univ., March. 2000.

[24] S. Noel, D. Wijesekera and C. Youman, "Modern intrusion detection, data mining, and degrees of attack guilt," in *Applications of Data Mining in Computer Security,* 2002, pp. 2-25.

[25] V. Kumar, J. Srivastava and A. Lazarević, " Intrusion detection: A survey," in *Managing Cyber Threats: Issues, Approaches, and Challenges*Anonymous Springer Verlag, 2005, .

[26] S. Kumar and E. H. Spafford, "An application of pattern matching in intrusion detection," 1994.

[27] D. S. Bauer and M. E. Koblentz, "NIDX-an expert system for real-time network intrusion detection," in *Computer Networking Symposium, 1988., Proceedings of the,* 2002, pp. 98-106.

[28] C. Dowell and P. Ramstedt, "The ComputerWatch data reduction tool," in 1990//, pp. 99.

[29] J. R. Winkler, "A UNIX prototype for intrusion and anomaly detection in secure networks," in *Standards - the Key to the Future,* 1990, pp. 115-24.

[30] M. Esmaili, R. Safavi-Naini and M. Bala Balachandran, "AUTOGUARD: A continuous case-based intrusion detection system," in *ACSC'97,* 1997, pp. 392-401.

[31] S. Northcutt, "Intrusion Detection Shadow Style," *SANS Institute,* 1999.

[32] U. Lindqvist and P. A. Porras, "Detecting computer and network misuse through the production-based expert system toolset (P-BEST)," in *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on,* 2002, pp. 146-161.

[33] T. F. Lunt, "IDES: An intelligent system for detecting intruders," in *Proceedings of the Symposium: Computer Security, Threat and Countermeasures,* 1990, .

[34] D. Anderson, T. Frivold, A. Tamaru, A. Valdes and B. Release, "Next Generation Intrusion Detection Expert System (NIDES), Software Users Manual," 1994.

[35] P. A. Porras and P. G. Neumann, "EMERALD: Event monitoring enabling responses to anomalous live disturbances," in *In Proceedings of the 20th National Information Systems Security Conference,* 1997, pp. 353-365.

[36] M. Roesch, "Snort-lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX Conference on System Administration,* 1999, pp. 229–238.

[37] M. J. Ranum, K. Landfield, M. Stolarchuk, M. Sienkiewicz, A. Lambeth and E. Wall, "Implementing a generalized tool for network monitoring* 1," *Information Security Technical Report,* vol. 3, pp. 53-64, 1998.

[38] T. Heberlein, "Network Security Monitor (NSM)-Final Report," *UC Davis: February,* 1995.

[39] Cisco Systems, "Netranger intrusion detection system technical overview," 1998.

[40] Cisco Systems, "Cisco intrusion detection," vol. 2005, .

[41] Computer Security Technology Center Lawrence Livermore National Laboratory, "Network intrusion detector (NID)," vol. 1998, .

[42] Internet Security Systems (Now IBM Internet Security Systems), "RealSecure," vol. 2005, .

[43] Science Applications International Corporation (SAIC), "Computer misuse detection system (CMDS)," vol. 2004, .

[44] Symatec Corporation, "Symantec intruder alert," vol. 2005, .

[45] S. E. Smaha, "Haystack: An intrusion detection system," in *Fourth Aerospace Computer Security Applications Conference,* 1988, .

[46] K. Ilgun, R. A. Kemmerer and P. A. Porras, "State transition analysis: A rule-based intrusion detection approach," *Software Engineering, IEEE Transactions on,* vol. 21, pp. 181-199, 2002.

[47] K. Ilgun, "USTAT: A real-time intrusion detection system for UNIX," in *Research in Security and Privacy, 1993. Proceedings., 1993 IEEE Computer Society Symposium on,* 2002, pp. 16-28.

[48] G. Vigna and R. A. Kemmerer, "NetSTAT: A network-based intrusion detection approach," in *Computer Security Applications Conference, 1998, Proceedings., 14th Annual,* 2002, pp. 25-34.

[49] M. Gao and M. C. Zhou, "Fuzzy intrusion detection based on fuzzy reasoning petri nets," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on,* 2003, pp. 1272-1277.

[50] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," in *Proceedings of the 7th Conference on USENIX Security Symposium-Volume 7,* 1998, pp. 6.

[51] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the JAM project," in *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings,* 2002, pp. 130-144.

[52] U. Zurutuza, "Sistemas de detección de intrusos. Estado del arte. State of the Art e-book," *CRIPTORED,* 2004.

[53] H. Li and K. Yamanishi, "Topic analysis using a finite mixture model* 1," *Information Processing & Management,* vol. 39, pp. 521-541, 2003.

[54] N. Ye and Q. Chen, "An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems," *Qual. Reliab. Eng. Int.,* vol. 17, pp. 105-112, 2001.

[55] T. Dübendorfer, A. Wagner, T. Hossmann and B. Plattner, "Flow-level traffic analysis of the blaster and sobig worm outbreaks in an internet backbone," *Intrusion and Malware Detection and Vulnerability Assessment,* pp. 103-122, 2005.

[56] N. Ye, S. M. Emran, X. Li and Q. Chen, "Statistical process control for computer intrusion detection," in *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings,* 2002, pp. 3-14.

[57] W. Dumouchel and M. Schonlau, "A comparison of test statistics for computer intrusion detection based on principal components regression of transition probabilities," in *Proceedings of the 30th Symposium on the Interface: Computing Science and Statistics,* 1998, pp. 404-413.

[58] L. W. Chang, M. L. Shyu, S. C. Chen and K. Sarinnapakorn, "A Novel Anomaly Detection Scheme Based on Principal Component Classifier," 2003.

[59] Barbará Daniel, N. Wu and S. Jajodia, "Detecting novel network intrusions using bayes estimators," in *Proceedings of the First SIAM Conference on Data Mining,* 2001, .

[60] L. Ertöz, E. Eilertson, A. Lazarevic, P. Tan, P. Dokas, V. Kumar and J. Srivastava, "Detection and Summarization of Novel Network Attacks Using Data Mining," 2004.

[61] K. Sequeira and M. Zaki, "ADMIT: Anomaly-based data mining for intrusions," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2002, pp. 386-395.

[62] T. D. Lane, "Machine learning techniques for the computer security domain of anomaly detection," *ETD Collection for Purdue University,* 2000.

[63] I. Balepin, S. Maltsev, J. Rowe and K. Levitt, "Using specification-based intrusion detection for automated response," in *Recent Advances in Intrusion Detection,* 2003, pp. 136-154.

[64] S. Forrest, S. A. Hofmeyr, A. Somayaji and T. A. Longstaff, "Sense of self for unix processes," in *Proceedings of the 1996 17th IEEE Symposium on Security and Privacy, may 6, 1996 - may 8,* 1996, pp. 120-128.

[65] A. Somayaji and S. Forrest, "Automated response using system-call delays," in *Proceedings of the 9th Conference on USENIX Security Symposium-Volume 9,* 2000, pp. 14.

[66] T. Toth and C. Kruegel, "Evaluating the impact of automated intrusion response mechanisms," in *Computer Security Applications Conference, 2002. Proceedings. 18th Annual,* 2003, pp. 301-310.

[67] W. Leea, W. Fanb, M. Miller, S. J. Stolfoc and E. Zadok, "Toward cost-sensitive modeling for intrusion detection and response," *Journal of Computer Security,* vol. 10, pp. 5-22, 2002.

[68] I. Krontiris, T. Dimitriou and F. C. Freiling, "Towards intrusion detection in wireless sensor networks," in *13th European Wireless Conference,* Paris, 2007, .

[69] I. Krontiris, T. Giannetsos and T. Dimitriou, "LIDeA: A distributed lightweight intrusion detection architecture for sensor networks," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks,* 2008, pp. 1-10.

[70] I. Krontiris, Z. Benenson, T. Giannetsos, F. Freiling and T. Dimitriou, "Cooperative intrusion detection in wireless sensor networks," *Wireless Sensor Networks,* pp. 263-278, 2009.

[71] A. P. R. da Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks,* 2005, pp. 16-23.

[72] P. Techateerawat and A. Jennings, "Energy efficiency of intrusion detection systems in wireless sensor networks," in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology,* 2006, pp. 227-230.

[73] A. Giannetsos, "Intrusion detection in wireless sensor networks," 2009.

[74] I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks," in *Wireless and Mobile Computing, Networking and Communications, 2005.(WiMob'2005), IEEE International Conference on,* 2005, pp. 253-259.

[75] F. Kargl, A. Klenk, M. Weber and S. Schlott, "Sensors for detection of misbehaving nodes in MANETs," in *In Proceedings of Workshop Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2004) (to Appear,* 2004, .

[76] L. Chong Eik, N. Mun Yong, L. Christopher and P. Marimuthu, "Intrusion Detection for Routing Attacks in Sensor Networks," *International Journal of Distributed Sensor Networks,* vol. 2, pp. 313-332, 2006.

[77] V. Bhuse and A. Gupta, "Anomaly intrusion detection in wireless sensor networks," *Journal of High Speed Networks,* vol. 15, pp. 33-51, 2006.

[78] I. Krontiris, T. Dimitriou, T. Giannetsos and M. Mpasoukos, "Intrusion detection of sinkhole attacks in wireless sensor networks," *Algorithmic Aspects of Wireless Sensor Networks,* pp. 150-161, 2008.

[79] A. Mishra, K. Nadkarni and A. Patcha, "Intrusion detection in wireless ad hoc networks," *Wireless Communications, IEEE,* vol. 11, pp. 48-60, 2005.

[80] K. Q. Yan, S. C. Wang and C. W. Liu, "A Hybrid Intrusion Detection System of Cluster-based Wireless Sensor Networks," .

[81] T. H. Hai, F. Khan and E. N. Huh, "Hybrid intrusion detection system for wireless sensor networks," in *Proceedings of the 2007 International Conference on Computational Science and its Applications-Volume Part II,* 2007, pp. 383-396.

[82] Anjum, F. (2007). Location dependent key management in sensor networks without using deployment knowledge., (pp. 1-10).

[83] Chan, H., Perrig, A., & Song, D. (2003). Random key predistribution schemes for sensor networks., (pp. 197-213).

[84] Du, W., Deng, J., Han, Y. S., Chen, S., & Varshney, P. K. (2004). A key management scheme for wireless sensor networks using deployment knowledge., 1.

[85] Eltoweissy, M., Moharrum, M., & Mukkamala, R. (2006). Dynamic key management in sensor networks. #IEEE_M_COM# , 44 (4), 122-130.

[86] Eschenauer, L., & Gligor, V. D. (2002). A key-management scheme for distributed sensor networks. (pp. 41-47). ACM.

[87] Huang, D., 0003, M. M., Medhi, D., & Harn, L. (2004). Location-aware key management scheme for wireless sensor networks., (pp. 29-42).

[88] Lamport, L. (1981). Password authentication with insecure communication. Commun. ACM , 24 (11), 770-772.

[89] Liu, D., Ning, P., & Li, R. (2005). Establishing pairwise keys in distributed sensor networks. ACM Trans. Inf. Syst. Secur. , 8 (1), 41-77.

[90] Perrig, A., Canetti, R., Tygar, J. D., & Song, D. (2000). Efficient authentication and signing of multicast streams over lossy channels., (pp. 56-73).

[91] Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., & Culler, D. E. (2002). SPINS: security protocols for sensor networks. Wirel. Netw. , 8 (5), 521-534.

[92] pSHIELD Consortium. (2011). SPD Self-X and Cryptographic Technologies.

[93] Xiao, Y., Rayi, V. K., Sun, B., Du, X., Hu, F., & Galloway, M. (2007). A survey of key management schemes in wireless sensor networks. Comput. Commun. , 30 (11-12), 2314-2341.

[94] Younis, M. F., Ghumman, K., & Eltoweissy, M. (2006). Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks. #IEEE_J_PDS# , 17 (8), 865-882.

[95] Zhu, S., Setia, S., & Jajodia, S. (2003). LEAP: efficient security mechanisms for large-scale distributed sensor networks. (pp. 62-72). ACM.

[96] Z. Yu and J. J. P. Tsai, "A framework of machine learning based intrusion detection for wireless sensor networks," in 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '08), 2008, pp. 272-9.

[97] F. Liu, X. Cheng and D. Chen, "Insider attacker detection in wireless sensor networks," in IEEE INFOCOM 2007, 2007, pp. 1936-44.

[98] J. Newsome, E. Shi, D. Song and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis & Defenses," 2004.

[99] DAMASIO A. (2000) The feeling of what happens - body, emotion and the rise of consciousness, Harvest Books.

[100] BRDICZKA O., P.C. CHEN, S. ZAIDENBERG, P. REIGNIER and J.L. CROWLEY (2006) Automatic Acquisition of Context Models and its Application to Video Surveillance, International Conference in Pattern Recognition, 1175-1178, DOI:10.1109/ICPR.2006.292.

[101] MARCENARO L., OBERTI F., FORESTI G., REGAZZONI C. (2001) Distributed architectures and logical-task decomposition in multimedia surveillance systems. Proceedings of the IEEE (10) (October 2001) 1419-1440

[102] MONCRIEFF S., S. VENKATESH e G. WEST (2008) Context aware privacy in visual surveillance, International Conference in Pattern Recognition, 1-4, DOI:10.1109/ICPR.2008.4761616.

[103] REMAGNINO P. e G.L. FORESTI (2005) Ambient Intelligence: A New Multidisciplinary Paradigm, Machine Vision and Applications, IEEE Transactions on Systems, Man and Cybernetics - Part A, 35, 1-6, DOI:10.1109/TSMCA.2004.838456.

[104] VELASTIN S., L. KHOUDOUR, B.P.L. LO, J. SUN and M.A. VICENCIO-SILVA (2004) PRISMATICA: A multi-sensor surveillance system for public transport network, 12th IEE Road Transport Information and Control Conference, 19-25.

[105] G. P. Perrucci, F. H. P. Fitzek, G. Sasso, W. Kellerer and J. Widmer, "On the impact of 2g and 3g network usage for mobile phones' battery life," in Wireless Conference, 2009. EW 2009. European, 2009, pp. 255-259.